NASA CR-177903

CSC/SD-81/6012

GODDARD MISSION ANALYSIS SYSTEM (GMAS) PRIMER

(NASA-CR-177903) GODDARD MISSION ANALYSIS SYSTEM (GMAS) PRIMER (Computer Sciences Corp.) 177 p

N87-70216

46802

Unclas

00/61 42925

Prepared For

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

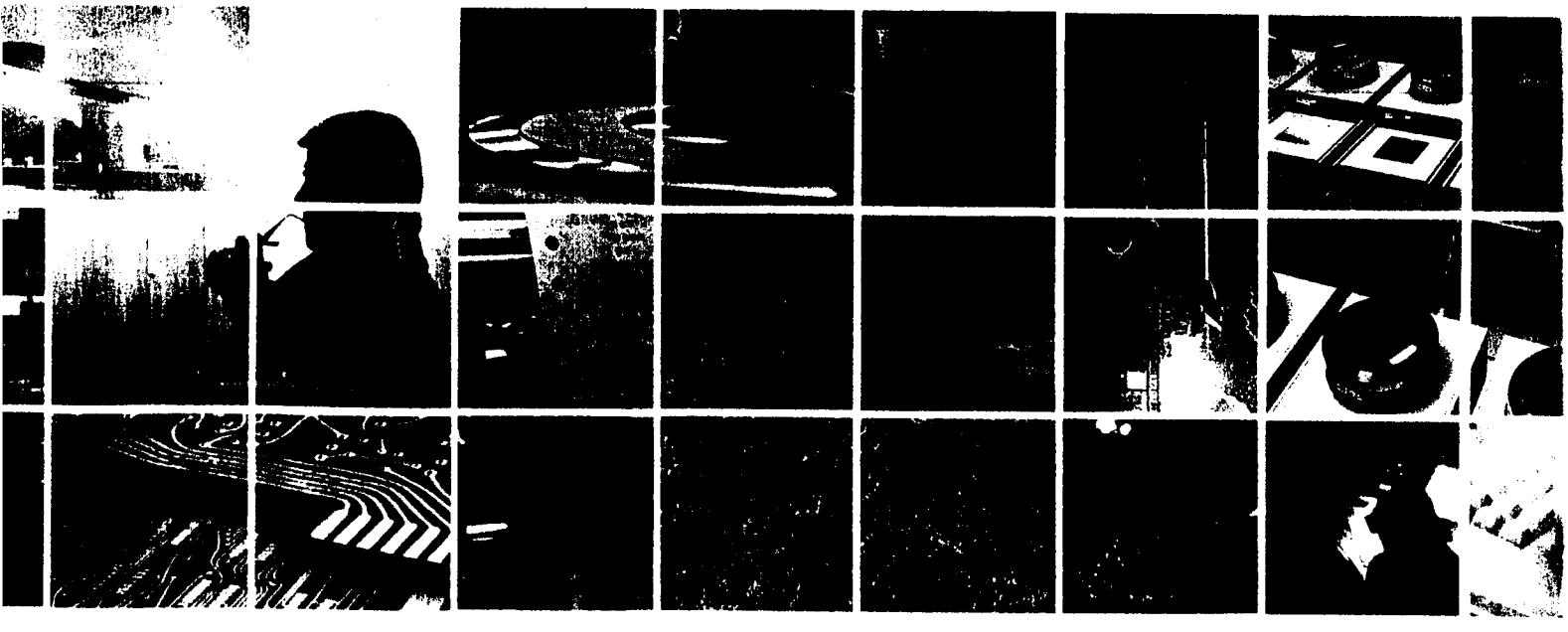
Goddard Space Flight Center

Greenbelt, Maryland

CONTRACT NAS 5-24300

Task Assignment 905

MAY 1981



CSC

COMPUTER SCIENCES CORPORATION

GODDARD MISSION ANALYSIS SYSTEM

(GMAS) PRIMER

Prepared for

GODDARD SPACE FLIGHT CENTER

By

COMPUTER SCIENCES CORPORATION

Under

Contract NAS 5-24300
Task Assignment 905

Prepared by:

R. A. Preiss — Date 4/10/81

Approved by:

A. L. Green — Date 5/5/81
Section Manager

Quality Assured by:

G. A. Snyder — Date 5/5/81

R. D. Headrick — Date 5/5/81
Department Manager

ACKNOWLEDGMENTS

The author wishes to acknowledge F. E. McGarry, C. R. Newman, and J. S. Watson of Goddard Space Flight Center for their direction and guidance in the preparation of this document. The author also thanks G. A. Snyder, J. E. Fry, and A. L. Green of Computer Sciences Corporation (CSC) for their help and support and H. Shepard of CSC for her excellent assistance in the editing and production of the document. In addition, the author would like to acknowledge all the users of the Goddard Mission Analysis System (GMAS) who generously supplied valuable input that ultimately affected the style and content of this primer.

ABSTRACT

This primer provides an introduction to the Goddard Mission Analysis System (GMAS). The primary objective of this document is to familiarize the reader with the fundamental operational mechanics of GMAS. The treatment of the subject matter is kept at an elementary level. Therefore, this primer is by no means an exhaustive study of GMAS capabilities. Basic information about propagating orbits, creating utilities, and generating automatic sequences is presented in a tutorial fashion with illustrations, examples, and explanations.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

Section 3 (Cont'd)

TABLE OF CONTENTS (Cont'd)

TABLE OF CONTENTS (Cont'd)

LIST OF ILLUSTRATIONS

Figure

LIST OF TABLES

Table

SECTION 1 - INTRODUCTION

1.1 DOCUMENT PURPOSE AND USE

This primer provides an introduction to the Goddard Mission Analysis System (GMAS). The primary objective of this document is to familiarize the reader with the fundamental operational mechanics of GMAS. The reader is assumed to have a working knowledge of the basic principles of astrodynamics. For example, the text contains references to orbital elements, coordinate systems, and spacecraft dynamics without any special explanations.

This primer is intended to be comprehensible to persons with very little computer background. For this reason, allusions to the internal software workings of the system have been avoided. To avoid apparent ambiguity and confusion, this primer presents a very structured approach to the operation of GMAS. Although the methods described in this document may not be the most efficient or "clever" ways to solve particular problems, they are the most straightforward. The more experienced user of GMAS may become more pragmatic in his approach to a mission problem. He may discover that some of the rules so strictly adhered to in this document can be relaxed or even ignored. He may also discover ways to manipulate GMAS internal software that greatly facilitate the solution to his problem. This, however, is beyond the scope of this document. A comprehensive description of GMAS can be found in the GMAS System Description (Reference 1) or the GMAS User's Guide (Reference 2).

To use this primer most effectively, the reader must have the following documents: (1) the GMAS User's Guide (Reference 2) and (2) the GMAS Software Resources document (Reference 3) and its update supplements (References 4 and 5). Many references are made to the contents of these documents

in this primer. The appendixes at the end of the User's Guide are crucial for any GMAS work.

Most of the content of this primer is supported by examples. To obtain the maximum benefit from this document, the reader should

1. Obtain the GMAS User's Guide and the GMAS Software Resources document to use in conjunction with the primer.

2. Read each section of the primer. Run the examples in each section on the computer. Determine whether the results obtained are the same as those specified in the primer.

3. Experiment by varying the input in the examples.

1.2 DOCUMENT OVERVIEW

This primer is composed of seven sections. The remainder of Section 1 provides some general information about GMAS. Section 2 describes a sample GMAS deck. Section 3 specifies the steps necessary to propagate a satellite orbit using GMAS. Section 4 describes the creation of an automatic sequence. Section 5 discusses the use of special GMAS capabilities in solving mission-related problems. Section 6 specifies the steps necessary to build a user module. Section 7 discusses various miscellaneous GMAS features, including automatic sequence libraries and the interactive mode.

1.3 GMAS OVERVIEW

1.3.1 GMAS DEFINED

GMAS is a collection of computer programs (software) that are currently stored on an IBM S/360 computer located in Building 3 at Goddard Space Flight Center (GSFC). Input to GMAS is usually punched on cards and is in the form of a

program (as opposed to data input). The GMAS input is called an automatic sequence. GMAS is primarily used to solve mission analysis problems.

1.3.2 GMAS CAPABILITIES

GMAS has a number of capabilities, including the following:

- GMAS can simulate satellite orbits with arbitrary orbital elements, force models, and stopping criteria. The user may choose from an assortment of propagators or design his own propagator to suit his mission-specific problems. Many different levels of orbital information printouts are available to the user. If the standard GMAS assortment of output parameters is not satisfactory, the user may develop his own output.

- GMAS can perform shadow and station coverage studies. It can create (or read) orbit (ORBIT) files and read ephemeris (EPHEM) files.

- GMAS has a graphing capability that can be used to generate comparison graphs of selected orbital parameters of two satellites.

- GMAS has a targeting and optimization capability that can be used by mission analysts to target maneuvers.

- GMAS has a Monte Carlo error analysis package for performing statistical studies.

The most useful aspects of GMAS are its flexibility and versatility. With the input capability (automatic sequences), the user can string together any collection of stored routines (utilities) in any desired configuration. These utilities can also communicate through special communication variables (dynamic arrays). The combined use of the GMAS utilities and user-provided utilities makes the capabilities of GMAS virtually unlimited.

1.3.3 GMAS ADVANTAGES

Some of the advantages of using GMAS are as follows:

- Software development time savings--Many other software systems (e.g., the Research and Development Goddard Trajectory Determination System (R&D GTDS)) are hard coded. For such systems, extensive software work is often necessary even for a relatively small alteration. GMAS is modularized; thus the user can concern himself with the creation or enhancement of a particular module without having to worry about the effects it will have on the entire system.

- Core savings and efficiency--Through modularization, GMAS is able to operate much more efficiently than its cumbersome counterparts, which may be laden with unused or dead code. Since GMAS programs require less computer memory (core), their turnaround time is more rapid.

- Stability--Because of its modularization, GMAS is a very stable system. It does not suffer from having "too many chefs in the kitchen." Each user creates his own utilities and stores them in his own private library. The system itself therefore requires little maintenance. System enhancements are developed by a small, coordinated group.

- Size--The effective size of GMAS is limited only by the size of the user's utility library. The user can do his library work on timesharing option (TSO) and make background runs to test his utilities.

- Flexibility--Under control of the automatic sequence, GMAS can perform a number of tasks (utilities), transferring information from one task to another. These tasks can be arranged in different orders, and sequences of tasks can be performed until certain conditions are met. In large, hard-coded, inflexible systems, software work must be

done for each described arrangement. In smaller, more flexible systems, the user is often forced to transfer information manually, which may be detrimental to efficiency and accuracy.

SECTION 2 - SAMPLE GMAS DECK

GMAS can be run on the IBM S/360 computer with a deck of cards (batch mode). Figure 2-1 shows a deck of Hollerith cards that can be used to simulate a satellite orbit with GMAS. Each Hollerith card has 80 columns. These cards can be typed (punched) on a keypunch.

Section 2.1 states Problem 1, a mission analysis problem. Section 2.2 describes the card deck that can be used to solve this problem. Section 2.3 provides examples of the printouts resulting from the use of this deck.

2.1 PROBLEM 1

Given the following, propagate the orbit of a satellite for 1 day and print out the default information, which includes the Keplerian and Cartesian coordinates:[1]

Epoch:	July 29, 1980; 07:09:58.4 Greenwich mean time (GMT)
Elements:	Cartesian: x = 7000.0 kilometers y = 200.0 kilometers z = 300.0 kilometers $\dot{x}$ = 0.0 kilometers per second $\dot{y}$ = 3.3 kilometers per second $\dot{z}$ = 8.0 kilometers per second
Coordinate system of input:	True Earth equator and equinox of date
Propagator:	Fixed-step Cowell (step size = 100 seconds)
Force model:	4-by-4 Earth field, no drag or solar radiation pressure, effects of Moon and Sun included (default)

[1]Default output is called level 2 output. Table C-1d of the User's Guide provides a complete description of level 2 output.

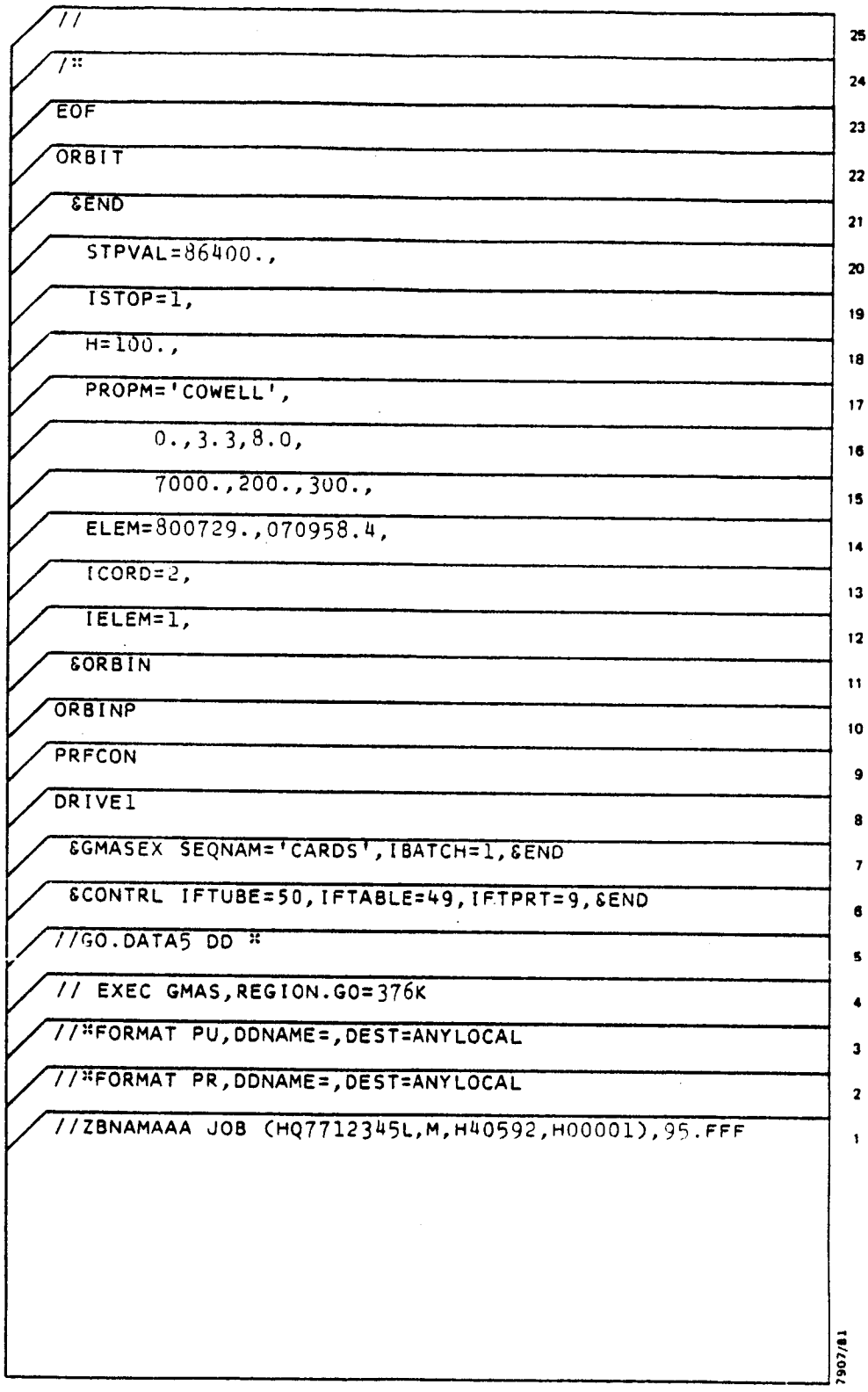


Figure 2-1. Sample GMAS Deck

2.2 CARD DECK FOR PROBLEM 1

The deck of cards illustrated in Figure 2-1 can be used to solve Problem 1. This deck includes 25 cards: 5 initial job control language (JCL) cards, 18 GMAS input cards, and 2 final JCL cards. Each of these cards, numbered in text and in Figure 2-1 for convenience, is discussed below. In this discussion, the underlined portions of the card formats represent areas that may be changed from run to run at the user's discretion.

2.2.1 INITIAL JCL CARDS

First, the user must provide some cards that give the computer the information it needs to run his job. Cards that give the computer this information are called JCL cards. All JCL cards contain slashes (//) in their first two columns.

Each of the five initial JCL cards is described below.

- Card 1--The first card in the deck is called a JOB card. Its format is as follows:

```
col. 1
↓
//ZBNAMAAA JOB (HQ7712345L,M,H40592,H00001),95.FFF
```

user ID	job ID	accounting information	time limits	output box

ZBNAM is the user's identification (ID). This tells the computer who is running this job. If the user does not have an ID, he must apply for one.

AAA is the job ID. The user may use any three letters and/or numbers in these columns. (The user ID and the job ID appear at the top of the first page of computer printout that results from the computer run (see Section 2.3). If the user is making several runs, he can identify them quickly and easily by choosing appropriate job IDs.)

JOB tells the computer that this card is a JOB card.

HQ7712345L,M,H40592 is accounting information. These numbers are supplied to the user and remain constant unless some authorization change occurs.

H00001 specifies the time limits for the run. Computer time falls into two categories: central processing unit (CPU) and input/output (I/O). CPU time is the amount of time the computer spends processing the information it receives. I/O time is the amount of time the computer spends collecting and dispersing this information and printing out the results. In this example, the user has allowed a maximum of 1/2 minute (H00) of CPU time and 1 minute (001) of I/O time. If the program being run exceeds either of these limits, the computer will not complete the job, and the user will get the run back only partially completed with an error message. The easiest way for the user to determine what numbers should go in this area is to look at the bottom of the last job summary printout page of a similar run, where a summary of the CPU and I/O time used is provided (see Figure 2-2). The user should keep in mind that the computer gives shorter jobs priority over longer jobs. Therefore, it is unwise for a user to specify CPU-I/O requirements of, for example, 005005 if the job can be run in less than 001001.

95.FFF specifies the box in which the computer staff is to put the run printout.

More detailed information on the format of the JOB card can be found in Reference 6.

- Cards 2 and 3--The two JCL cards following the JOB card tell the computer on which device(s) it is to generate the printed and/or punched output. These two cards should be included in the deck if the output from the run is to be

```
IEF237I 452   ALLOCATED TO FT17F001
IEF237I 243   ALLOCATED TO FT18F001
IEF237I 489   ALLOCATED TO FT19F001
IEF237I 335   ALLOCATED TO FT20F001
IEF237I 335   ALLOCATED TO FT21F001
IEF237I 335   ALLOCATED TO FT22F001
IEF237I 243   ALLOCATED TO FT26F001
IEF237I 458   ALLOCATED TO FT29F001
IEF237I 335   ALLOCATED TO FT30F001
IEF237I 458   ALLOCATED TO FT38F001
IEF237I 489   ALLOCATED TO FT49F001
IEF237I 333   ALLOCATED TO GESSMSG
IEF237I 452   ALLOCATED TO
IEF237I 762   ALLOCATED TO SYSPRINT
IEC130I FT50F001 DD STATEMENT MISSING
IEF142I - STEP WAS EXECUTED - COND CODE 0000
IEF285I   GJMAS.GO.LOAD                                    KEPT          DDNAME=STEPLIB -1          0 EXCPS
IEF285I   VOL SER NOS= DISK17.
IEF285I   GJMAS.GO.LOAD                                    KEPT          DDNAME=STEPLIB -2          0 EXCPS
IEF285I   VOL SER NOS= DISK17.
IEF285I   GJMAS.UTILITY.DATA                               KEPT          DDNAME=FT01F001           15 EXCPS
IEF285I   VOL SER NOS= DISK05.
IEF285I   SYS81008.T171125.RV001. ZBNAMAAA .EXECFILE       DELETED       DDNAME=FT02F001            4 EXCPS
IEF285I   VOL SER NOS= SCR001.
IEF285I   SYS81008.T171125.RV001. ZBNAMAAA .WORK           DELETED       DDNAME=FT03F001            5 EXCPS
IEF285I   VOL SER NOS= SCR001.
IEF285I   GJMAS.AUTOSEQ.DATA                              KEPT          DDNAME=FT04F001            0 EXCPS
IEF285I   VOL SER NOS= DISK17.
IEF285I   SYS81008.T171125.RV001. ZBNAMAAA .ASP10001       DELETED       DDNAME=FT05F001            4 EXCPS
IEF285I   VOL SER NOS= 012648.
IEF285I   SYS81008.T171125.RV001. ZBNAMAAA .ASP0A001       DELETED       DDNAME=FT06F001            9 EXCPS
IEF285I   VOL SER NOS= ASP760.
IEF285I   SYS81008.T171125.RV001. ZBNAMAAA .DISPLAY        DELETED       DDNAME=FT08F001            1 EXCPS
IEF285I   VOL SER NOS= SCR001.
IEF285I   SYS81008.T171125.RV001. ZBNAMAAA .ASP0A002       DELETED       DDNAME=FT09F001           20 EXCPS
IEF285I   VOL SER NOS= ASP761.
IEF285I   GJMAS.DYNAMICS.DATA                              KEPT          DDNAME=FT10F001           18 EXCPS
IEF285I   VOL SER NOS= DISK09.
IEF285I   SYS81008.T171125.RV001. ZBNAMAAA .DYN            DELETED       DDNAME=FT11F001            0 EXCPS
IEF285I   VOL SER NOS= SCR001.
IEF285I   ORBIT.GTDS.SLP1950.DATA                          KEPT          DDNAME=FT14F001            3 EXCPS
IEF285I   VOL SER NOS= DISKD0.
IEF285I   ORBIT.GTDS.JACCHIA.DATA                          KEPT          DDNAME=FT15F001            0 EXCPS
IEF285I   VOL SER NOS= DISK19.
IEF285I   GJMAS.ERRMSG.DATA                                KEPT          DDNAME=FT17F001            0 EXCPS
IEF285I   VOL SER NOS= DISK02.
IEF285I   SYS81008.T171125.RV001. ZBNAMAAA .CURR           DELETED       DDNAME=FT18F001            1 EXCPS
IEF285I   VOL SER NOS= SCR001.
IEF285I   GJMAS.NEWS.DATA                                  KEPT          DDNAME=FT19F001            2 EXCPS
IEF285I   VOL SER NOS= DISK39.
IEF285I   ORBIT.GTDS.EARTHFLD.DATA                         KEPT          DDNAME=FT20F001            0 EXCPS
IEF285I   VOL SER NOS= DISK19.
IEF285I   ORBIT.GTDS.LUNARFLD.DATA                         KEPT          DDNAME=FT21F001            0 EXCPS
IEF285I   VOL SER NOS= DISK19.
IEF285I   ORBIT.GTDS.ATMOSDEN.DATA                         KEPT          DDNAME=FT22F001            0 EXCPS
IEF285I   VOL SER NOS= DISK19.
IEF285I   SYS81008.T171125.RV001. ZBNAMAAA .R0000333       DELETED       DDNAME=FT26F001            0 EXCPS
IEF285I   VOL SER NOS= SCR001.
IEF285I   GJMAS.STATION.DATA                               KEPT          DDNAME=FT29F001            0 EXCPS
IEF285I   VOL SER NOS= DISK17.
IEF285I   ORBIT.GTDS.D24HOUR.DATA                          KEPT          DDNAME=FT30F001            0 EXCPS
IEF285I   VOL SER NOS= DISK19.
IEF285I   ORBIT.GTDS.TIMCOF.DATA                           KEPT          DDNAME=FT38F001            4 EXCPS
IEF285I   VOL SER NOS= DISKD0.
IEF285I   GJMAS.NONRES.DATA                                KEPT          DDNAME=FT49F001            4 EXCPS
IEF285I   VOL SER NOS= DISK09.
IEF285I   ATTIT.GESSMSG.DATA                               KEPT          DDNAME=GESSMSG -1          4 EXCPS
IEF285I   VOL SER NOS= DISKS2.
IEF285I   GJMAS.GMASMSG.DATA                               KEPT          DDNAME=GESSMSG -2          0 EXCPS
IEF285I   VOL SER NOS= DISK02.
***
*** CHARGE TIME          CPU=****0.070 MIN   I/O=****0.180 MIN
*** I/O TIME IN SEC.     DRUM=*****0.86 DISK=*****0.50 CELL=*****0.00 TAPE=*****0.00 GRAF=*****0.00  ASP=*****0.49
*** I/O COUNTS           TOTAL-EXCPS=*****325     MISC-DEVICES=******00  CARDS=00000
*** REGION               START=1586K  SIZE=1376K   MAX-USED=1344K  PERCENT-USED=09%
*** ZBKAPAAA/GO          STEP=001  PGM=GMAS                        START=01-08-81/17.35.33  STOP=01-08-81/17.36.23
***
***
***
*** CHARGE TIME          CPU=****0.070 MIN   I/O=****0.180 MIN
*** I/O TIME IN SEC.     DRUM=*****0.86 DISK=*****0.50 CELL=*****0.00 TAPE=*****0.00 GRAF=*****0.00  ASP=*****0.49
*** I/O COUNTS           TOTAL-EXCPS=*****325     MISC-DEVICES=******00
*** ZBNAMAAA             ROM=J  CLASS=A  PRTY=08                   START=01-08-81/17.35.33  STOP=01-08-81/17.36.23
***
*** NASA/GSFC 360/95 GI      SYSTEM=MVT-21.8     FOR PROBLEM ASSISTANCE, CALL P&C ON 344-6768
AMDS09 JOB 2648 ( ZBNAMAAA ) IN BREAKDOWN
```

Figure 2-2. Computer-Printed Job Summary

directed to the center facility for printing. The formats of these cards are as follows:

```
col. 1
↓
//*FORMAT PR,DDNAME=,DEST=ANYLOCAL
//*FORMAT PU,DDNAME=,DEST=ANYLOCAL
```

- Card 4--The next JCL card tells the computer to run GMAS. The format of this card is as follows:

```
col. 1
↓
// EXEC GMAS,REGION.GO=376K
   ↑     ↑
   one blank space
```

REGION.GO tells the computer how much memory space (core) it will need to run GMAS. Since GMAS is flexible in this regard, it is helpful for the user to know the size of region required to run his application. If the user's application requires a larger region than he specifies on this card, the computer will not run the job. Most GMAS applications run in less than 400K bytes of core. To propagate a satellite orbit with a Cowell propagator requires only 376K bytes of core. The user must try to be accurate in estimating the core requirement. For the GMAS beginner, fixing the region at 376K is safe. A more experienced user, however, will vary the region size depending on his application. If REGION.GO is omitted, the GMAS default region of 400K is used. The easiest way for the user to determine the core size necessary for his particular run is to look at the bottom of the last job summary printout page of a similar run, where the core size used during the run is provided (see Figure 2-2).

- Card 5--The next JCL card introduces the GMAS input cards. Its format is as follows:

```
col. 1     one blank space
↓          ↓  ↓
//GO.DATA5 DD *
```

The format of this card always remains the same.

2.2.2 GMAS INPUT CARDS

The GMAS input cards in this deck include 2 NAMELIST cards and 16 automatic sequence cards.

2.2.2.1 NAMELIST Cards

The first two GMAS input cards (cards 6 and 7 of this deck) are used for controlling the execution of GMAS. The experienced user can vary these cards, called NAMELIST cards, to make use of certain GMAS capabilities. (These capabilities are discussed in subsequent sections.) For the basic operation of GMAS using a card deck, the user will not need to modify the two NAMELIST cards, whose formats are as follows:

```
col. 2
▼
&CONTRL IFTUBE=50,IFTABL=49,IFTPRT=9,&END
&GMASEX SEQNAM='CARDS',IBATCH=1,&END
```

2.2.2.2 Automatic Sequence Cards

The 16 automatic sequence cards (cards 8 through 23 of this deck) are as follows:

- Cards 8 and 9--The first two cards of the automatic sequence each begin in column 1. The first (card 8) tells GMAS to use the general-purpose driver, DRIVE1; the second (card 9) selects the profile controller, PRFCON, to control GMAS processing. All automatic sequences except those used for targeting or Monte Carlo error analysis start with these two cards, whose formats are as follows:

```
col. 1
▼
DRIVE1
PRFCON
```

- Card 10--The next card, which selects the orbit input processor (ORBINP), begins in column 1. ORBINP must be

used to enter information for an orbit propagation. The ORBINP card format is as follows:

```
col. 1
↓
ORBINP
```

- Card 11--The next card, called the &ORBIN card, must follow the ORBINP card and must begin in column 2. This card introduces a series of cards specifying input variables and their values. Its format is as follows:

```
col. 2
↓
&ORBIN
```

- Cards 12 through 20--The next nine cards in this deck contain input variables and their values, separated by commas. Appendix C of the User's Guide provides a complete list of the &ORBIN variables, their default values, and their definitions. The input deck only needs to include cards containing those variables whose values the user wisnes to change. In this example (see Section 2.1), the user has set the following input variables and values:

Variable/Value	Explanation
IELEM=1,	A value of 1 for variable IELEM indicates that the epoch elements are Cartesian
ICORD=2,	A value of 2 for variable ICORD indicates that the coordinate system is true Earth equator and equinox of date
ELEM=800729.,070958.4, 7000.,200.,300., 0.,3.3,8.0,	ELEM is an eight-element array. The first two positions contain the packed year, month, day and hour, minute, second of epoch in the form YYMMDD.,HHMMSS.S; when IELEM=1, the next six positions contain the position and velocity of the satellite at the specified epoch and in the coordinate system specified by variable ICORD

Variable/Value	Explanation
PROPM='COWELL',	This specifies that the Cowell propagator is to be used to propagate the orbit
H=100.,	This specifies that the step size of the propagator is 100 seconds
ISTOP=1,	A value of 1 for variable ISTOP indicates that propagation is to stop when a specified time is reached
STPVAL=86400.,	This specifies that propagation is to stop at 86,400.0 seconds (1 day) from epoch time

- Card 21--The &END card, which signifies the end of the orbit input information, must follow the last input variable card. The &END card must begin in column 2. Its format is as follows:

```
col. 2
↓
&END
```

- Card 22--The next card, the ORBIT card, tells GMAS to propagate an orbit using the information given under ORBINP and the default information. This card must begin in column 1. Its format is as follows:

```
col. 1
↓
ORBIT
```

- Card 23--The last card of the automatic sequence is the end-of-file (EOF) card. All automatic sequences must end with this card. This card must begin in column 1. Its format is as follows:

```
col. 1
↓
EOF
```

2.2.3 FINAL JCL CARDS

Two final JCL cards follow the GMAS input cards. These two cards are as follows:

- Card 24--The /* card indicates to the computer the end of the GMAS input cards. Its use is optional. When used, the card must begin in column 1. Its format is as follows:

```
col. 1
↓
/*
```

- Card 25--The last card of the deck, the // card, indicates to the computer the end of the user's job. This card is used to separate one user's job from another as they are read in with a card reader. All decks of cards submitted to the computer must start with a JOB card and end with a // card. The // card is usually flipped over before it is punched. This helps the person operating the card reader to distinguish between jobs.

2.3 RESULTING PRINTOUTS

Figures 2-3 through 2-9 show some of the computer printouts resulting from use of the deck shown in Figure 2-1.

Figure 2-3 shows the first page of computer printout. The bold letters on this printout specify the user and run IDs.

Figure 2-4 shows three pages of messages from the computer. The JCL input is printed on these pages along with a collection of system information that can be ignored by the computer novice. The computer usage summary located at the bottom of the last page of Figure 2-4 specifies the CPU and I/O time used by the run. In this example, the run required 0.070 minute of CPU time and 0.130 minute of I/O time. These times are important in determining the time limitations to be entered on the JOB card for future runs.

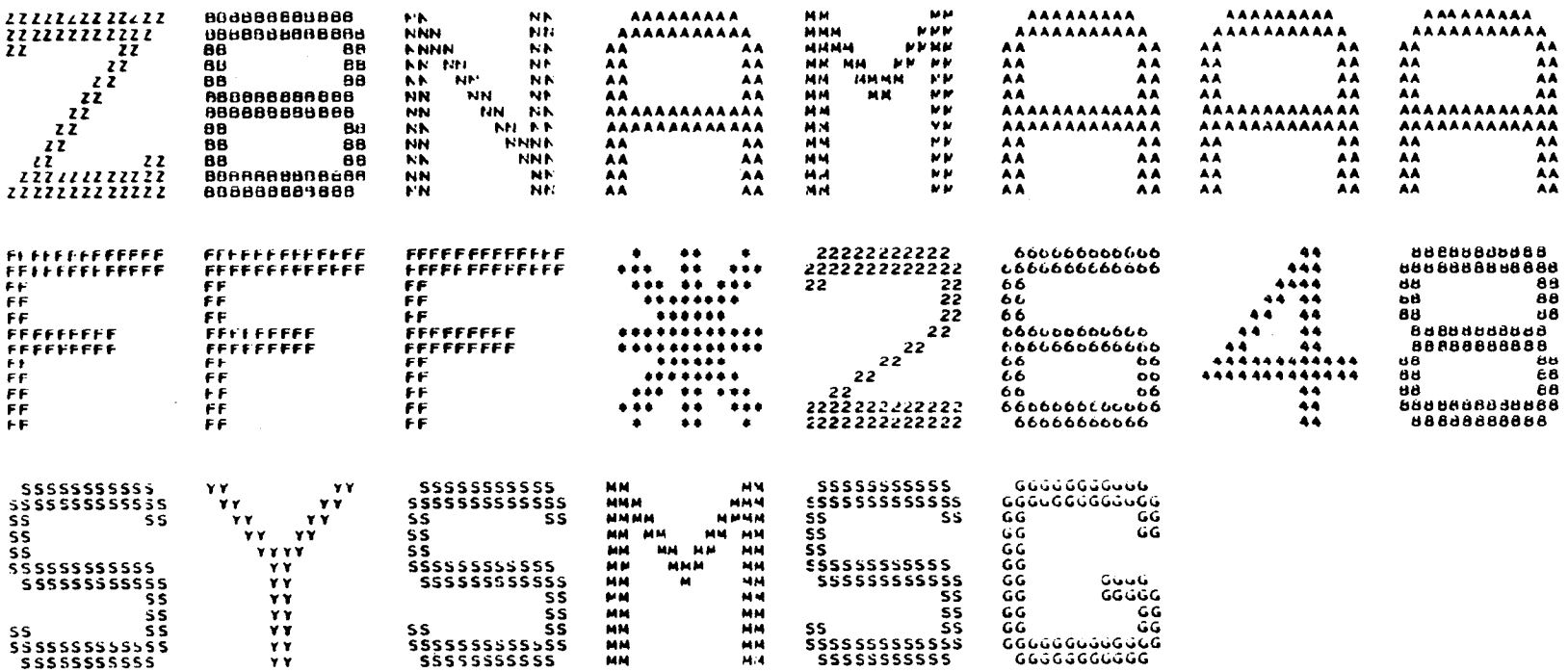


Figure 2-3. First Page of Computer Printout

```
ISV40 JOB CRIGIN FROM GROUP=CSC        . DSP=CR . DEVICE=GSFC5RD1, 042
.ZBNAMAAA JOB ([illegible]),95,FFF,MSGLEVEL=1
//*FORMAT PR,DDNAME=,DEST=ANYLOCAL
//*FORMAT PU,DDNAME=,DEST=ANYLOCAL
// EXEC GMAS,REGION.GO=375K
//GO.DATAS DD *
/*
//

 LOCATE' 2648GJMAS.GO.LOAD                                                 .
 AL 26480E001/DISK170004
 LOCATE' 2648GJMAS.GO.LOAD                                                 .
 AL 26480E001/DISK170004
 LOCATE' 2648GJMAS.UTILITY.DATA                                            .
 AL 26480E001/DISK050004
 LOCATE' 2648GJMAS.AUTOSEQ.DATA                                            .
 AL 26480E001/DISK170004
 LOCATE' 2648GJMAS.DYNAMICS.DATA                                           .
 AL 26480E001/DISK090004
 LOCATE' 2648ORBIT.GTDS.SLP1950.DATA                                       .
 AL 26480E001/DISK000004
 LOCATE' 2648ORBIT.GTDS.JACCHIA.DATA                                       .
 AL 26480E001/DISK190004
 LOCATE' 2648GJMAS.ERRMSG.DATA                                             .
 AL 26480E001/DISK020004
 LOCATE' 2648GJMAS.NEWS.DATA                                               .
 AL 26480E001/DISK090004
 LOCATE' 2648ORBIT.GTDS.EARTHFLD.DATA                                      .
 AL 26480E001/DISK190004
 LOCATE' 2648ORBIT.GTDS.LUNARFLD.DATA                                      .
 AL 26480E001/DISK190004
 LOCATE' 2648ORBIT.GTDS.ATMOSDEN.DATA                                      .
 AL 26480E001/DISK190004
 LOCATE' 2648GJMAS.STATION.DATA                                            .
 AL 26480E001/DISK170004
 LOCATE' 2648ORBIT.GTDS.D24HOUR.DATA                                       .
 AL 26480E001/DISK190004
 LOCATE' 2648ORBIT.GTDS.TIMCOF.DATA                                        .
 AL 26480E001/DISK000004
 LOCATE' 2648GJMAS.NONRES.DATA                                             .
 AL 26480E001/DISK090004
 LOCATE' 2648ATTIT.GESSMSG.DATA                                            .
 AL 26480E001/DISKS20004
 LOCATE' 2648GJMAS.GMASMSG.DATA                                            .
 AL 26480E001/DISK020004

AMDS01 JOB 2648  (ZBNAMAAA)  IN SETUP ON MAIN=G1
AMDS02 STEPLIB   ZBNAMAAA  USING D # DISK17 ON 45E
AMDS02 FT01F001  ZBNAMAAA  USING D # DISK05 ON 455
AMDS02 FT10F001  ZBNAMAAA  USING D # DISK09 ON 459
AMDS02 FT14F001  ZBNAMAAA  USING D # DISK03 ON 45E
AMDS02 FT15F001  ZBNAMAAA  USING D # DISK19 ON 335
AMDS02 FT17F001  ZBNAMAAA  USING D # DISK02 ON 452
AMDS02 GESSMSG   ZBNAMAAA  USING D # DISKS2 ON 333
  ZBNAMAAA   IEF403I  ZBNAMAAA  STARTED     TIME=17.35.34
  ZBNAMAAA   IEF234E D 761,ASP761
* ZBNAMAAA *02 IECASPO 760 IS ZBNAMAAA  A          GO        FT06F001
* ZBNAMAAA *02 IECASPO 75F IS ZBNAMAAA             GO        ASPI0001
* ZBNAMAAA *02 IECASPO 761 IS ZBNAMAAA  A          GO        FT09F001
* ZBNAMAAA *02 IECASPO 75F IS ZBNAMAAA             GO        ASPI0001
  ZBNAMAAA   IEC202E K 75F,012648,NL, ZBNAMAAA .GO
  ZBNAMAAA   IEF404I  ZBNAMAAA  ENDED       TIME=17.36.23
//ZBNAMAAA   JOB (GJ0021311T,T,G00405,001001),95,FFF,MSGLEVEL=1          FFF  JJ
// EXEC GMAS,REGION.GO=375K
XXGMAS PROC                                                              00000100
XXGO EXEC PGM=GMAS,REGION=400K                                           00000200
XXSTEPLIB  DD DSN=GJMAS.GO.LOAD,DISP=SHR,DCB=BUFNO=1                     00000300
XX   DD DSN=GJMAS.GO.LOAD,DISP=SHR,DCB=BUFNO=1                           00000400
XXFT01F001 DD DISP=SHR,DCB=BUFNO=1,                DEFAULT DATA FILE     00000500
XX             DSN=GJMAS.UTILITY.DATA                                    00000600
XXFT02F001 DD DISP=(NEW,DELETE),  UPDATED AUTO SEQUENCE STATEMENTS       00000700
XX            DCB=(RECFM=FBS,LRECL=64,BLKSIZE=3456,BUFNO=1),             00000800
XX             SPACE=(TRK,(2,1)),DSN=&EXECFILE,UNIT=DISK                 00000900
XXFT03F001 DD DISP=(NEW,DELETE),  SCRATCH/WORKING FILE                   00001000
XX             DSN=&WORK,UNIT=DISK,SPACE=(TRK,(2,1)),                    00001100
XX            DCB=(RECFM=FBS,LRECL=64,BLKSIZE=3456,BUFNO=1)              00001200
XXFT04F001 DD DISP=SHR,                AUTOMATIC SEQUENCE FILE           00001300
XX             DSN=GJMAS.AUTOSEQ.DATA,DCB=BUFNO=1                        00001400
XXFT05F001 DD DDNAME=DATAS             DATA CARD INPUT                   00001500
XXFT06F001 DD SYSOUT=A,                PRINTER OUTPUT                    00001600
```

Figure 2-4. Messages From the Computer (1 of 3)

```
XX             DCB=(RECFM=V6A,LRECL=137,BLKSIZE=1922,BUFNO=1)                    00001700
XXGO.FT08F001  DD DISP=(NEW,DELETE),      SCRATCH FOR TARGETING DISPLAYS         00001800
XX        DCB=(RECFM=FBS,LRECL=72,BLKSIZE=3456,BUFNO=1),                         00001900
XX        DSN=&&DISPLAY,UNIT=DISK,SPACE=(CYL,(2,2))                              00002000
XXPT09F001 DD SYSOUT=A,                   GESS PRINTER OUTPUT                    00002100
XX             DCB=(RECFM=V6A,LRECL=137,BLKSIZE=1922,BUFNO=1)                    00002200
XXFT10F001 DD DISP=SHR,                   DYNAMICS FILE                          00002300
XX             DSN=GJMAS.DYNAMICS.DATA,DCB=BUFNO=1                               00002400
XXFT11F001 DD DISP=(,DELETE),             UPDATED DYNAMICS FILE (OPTIONAL)       00002500
XX             DSN=&DYN,UNIT=2314,SPACE=(TRK,3),DCB=BUFNO=1                      00002600
XXFT12F001 DD DUMMY,                      TAPE ORBIT FILE WITHOUT PARTIALS       00002700
XX             UNIT=9TRACK,DCB=(RECFM=VS,LRECL=1096,BLKSIZE=1100,                00002800
XX             BUFNO=1),LABEL=(,SLP),DISP=SHR                                    00002900
XXFT13F001 DD DUMMY,                      TAPE ORBIT FILE WITH PARTIALS          00003000
XX             UNIT=9TRACK,DCB=(RECFM=VS,LRECL=6664,BLKSIZE=6668,                00003100
XX             BUFNO=1),LABEL=(,SLP),DISP=SHR                                    00003200
XXFT14F001 DD DISP=SHR,                   SOLAR/LUNAR/PLANETARY EPHEMERIS FILE   00003300
XX             DSN=ORBIT.GTDS.SLP1950.DATA,DCB=BUFNO=1                           00003400
XXFT15F001 DD DISP=SHR,                   JACCHIA ROBERTS ATMOSPHERIC FILE       00003500
XX             DSN=ORBIT.GTDS.JACCHIA.DATA,DCB=BUFNO=1                           00003600
XXFT16F001 DD DUMMY                       ERROR MESSAGE DATA SET FOR SCOPE       00003700
XXFT17F001 DD DISP=SHR,DSN=GJMAS.ERRMSG.DATA,DCB=BUFNO=1                        00003800
XXFT18F001 DD DISP=(NEW,DELETE),          CURRENT AUTO SEQ FILE                  00003900
XX             DSN=&CURR,UNIT=DISK,SPACE=(TRK,(2,1)),                            00004000
XX             DCB=(RECFM=FBS,LRECL=64,BLKSIZE=3456,BUFNO=1)                     00004100
XXFT19F001 DD DISP=SHR,          GMAS SYSTEM NEWS FILE                          00004200
XX             DSN=GJMAS.NEWS.DATA,DCB=BUFNO=1                                   00004300
XXFT20F001 DD DISP=SHR,                   EARTH POTENTIAL FIELD                  00004400
XX             DSN=ORBIT.GTDS.EARTHFLD.DATA,DCB=BUFNO=1                          00004500
XXFT21F001 DD DISP=SHR,                   LUNAR POTENTIAL FIELD                  00004600
XX             DSN=ORBIT.GTDS.LUNARFLD.DATA,DCB=BUFNO=1                          00004700
XXFT22F001 DD DISP=SHR,                   ATMOSPHERIC DENSITY -FILE              00004800
XX             DSN=ORBIT.GTDS.ATMOSDEN.DATA,DCB=BUFNO=1                          00004900
XXFT23F001 DD DUMMY,                      COMPARE SEQ ORBIT FILE 2. W/O PART     00005000
XX             UNIT=9TRACK,DCB=(RECFM=VS,LRECL=1096,BLKSIZE=1100,                00005100
XX             BUFNO=1),LABEL=(,SLP),DISP=SHR                                    00005200
XXFT24F001 DD DUMMY,                      COMPARE SEQ ORBIT FILE 2. WITH PAR     00005300
XX             UNIT=9TRACK,DCB=(RECFM=VS,LRECL=6664,BLKSIZE=6668,                00005400
XX             BUFNO=1),LABEL=(,SLP),DISP=SHR                                    00005500
XXFT25F001 DD DUMMY,                      DISK ORBIT FILE WITHOUT PARTIALS       00005600
XX             UNIT=DISK,DCB=(RECFM=F,BLKSIZE=1092,DSORG=DA,BUFNO=1),            00005700
XX             SPACE=(CYL,4)                                                     00005800
XXFT26F001 DD UNIT=DISK,                  DISK ORBIT FILE WITH PARTIALS          00005900
XX             DCB=(RECFM=F,BLKSIZE=6660,DSORG=DA,BUFNO=1),                      00006000
XX             SPACE=(CYL,12)                                                    00006100
XXFT27F001 DD DUMMY,                      COMPARE DA ORBIT FILE 2. W/O PARTIA    00006200
XX             UNIT=DISK,DCB=(RECFM=F,BLKSIZE=1092,DSORG=DA,BUFNO=1),            00006300
XX             SPACE=(CYL,2)                                                     00006400
XXFT28F001 DD DUMMY,                      COMPARE DA ORBIT FILE 2. WITH PART     00006500
XX             UNIT=DISK,DCB=(RECFM=F,BLKSIZE=6660,DSORG=DA,BUFNO=1),            00006600
XX             SPACE=(CYL,12)                                                    00006700
XXFT29F001 DD DISP=SHR,          SHORT STATION FILE                             00006800
XX             DSN=GJMAS.STATION.DATA,DCB=BUFNO=1                                00006900
XXFT30F001 DD DISP=SHR,                   24-HOUR HOLD FILE                      00007000
XX             DSN=ORBIT.GTDS.024HOUR.DATA,DCB=BUFNO=1                           00007100
XXFT38F001 DD DISP=SHR,                   TIME CONVERSION COEFFICIENTS FILE      00007200
XX             DSN=ORBIT.GTDS.TIMCOF.DATA,DCB=BUFNO=1                            00007300
XXFT49F001 DD DISP=SHR,                   STORED GESS DISPLAYS (OPTIONAL)        00007400
XX             DSN=GJMAS.NONRES.DATA                                             00007500
XXGESSMSG  DD DISP=SHR,                   GESS -MESSAGES                         00007600
XX             DSN=ATTIT.GESSMSG.DATA,DCB=BUFNO=1                                00007700
XX  DD DISP=SHR,DSN=GJMAS.GMASMSG.DATA,DCB=BUFNO=1    GMAS-MESSAGES           00007800
XXSYSPRINT DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=963,BUFNO=1)            00007900
//GO.DATA8 DD UNIT=(CTC,,DEFER),DISP=(OLD,DELETE),                             *
//  DSN=&&ASP10001,VOL=SER=012648,DCB=(RECFM=FB,LRECL=080,BLKSIZE=2000,       *  ASPGEN
//   BUFNO=1)                                                                    ASPGEN
//
IEF236I ALLOC. FOR  ZBNAMAAA  GO
IEF237I 458   ALLOCATED TO STEPLIB
IEF237I 458   ALLOCATED TO
IEF237I 455   ALLOCATED TO FT01F001
IEF237I 243   ALLOCATED TO FT02F001
IEF237I 243   ALLOCATED TO FT03F001
IEF237I 458   ALLOCATED TO FT04F001
IEF237I 75F   ALLOCATED TO FT05F001
IEF237I 760   ALLOCATED TO FT06F001
IEF237I 243   ALLOCATED TO FT08F001
IEF237I 761   ALLOCATED TO FT09F001
IEF237I 459   ALLOCATED TO FT10F001
IEF237I 243   ALLOCATED TO FT11F001
IEF237I 45E   ALLOCATED TO FT14F001
IEF237I 335   ALLOCATED TO FT15F001
```

Figure 2-4. Messages From the Computer (2 of 3)

```
IEF237I 452   ALLOCATED TO FT17F001
IEF237I 243   ALLOCATED TO FT18F001
IEF237I 489   ALLOCATED TO FT19F001
IEF237I 335   ALLOCATED TO FT20F001
IEF237I 335   ALLOCATED TO FT21F001
IEF237I 335   ALLOCATED TO FT22F001
IEF237I 243   ALLOCATED TO FT26F001
IEF237I 488   ALLOCATED TO FT29F001
IEF237I 339   ALLOCATED TO FT30F001
IEF237I 48E   ALLOCATED TO FT38F001
IEF237I 489   ALLOCATED TO FT49F001
IEF237I 333   ALLOCATED TO GESSMSG
IEF237I 482   ALLOCATED TO
IEF237I 762   ALLOCATED TO SYSPRINT
IEC130I FT50F001 DD STATEMENT MISSING
IEF142I - STEP WAS EXECUTED - COND CODE 0000
IEF285I   GJMAS.GO.LOAD                                  KEPT             DDNAME=STEPLIB -1          0 EXCPS
IEF285I   VOL SER NOS= DISK17.
IEF285I   GJMAS.GO.LOAD                                  KEPT             DDNAME=STEPLIB -2          0 EXCPS
IEF285I   VOL SER NOS= DISK17.
IEF285I   GJMAS.UTILITY.DATA                             KEPT             DDNAME=FT01F001           15 EXCPS
IEF285I   VOL SER NOS= DISK09.
IEF285I   SYS81008.T171125.RV001. ZBNAMAAA .EXECFILE     DELETED          DDNAME=FT02F001            4 EXCPS
IEF285I   VOL SER NOS= SCR001.
IEF285I   SYS81008.T171125.RV001. ZBNAMAAA .WORK         DELETED          DDNAME=FT03F001            5 EXCPS
IEF285I   VOL SER NOS= SCR001.
IEF285I   GJMAS.AUTOSEQ.DATA                             KEPT             DDNAME=FT04F001            0 EXCPS
IEF285I   VOL SER NOS= DISK17.
IEF285I   SYS81008.T171125.RV001. ZBNAMAAA .ASP10001     DELETED          DDNAME=FT05F001            4 EXCPS
IEF285I   VOL SER NOS= 012648.
IEF285I   SYS81008.T171125.RV001. ZBNAMAAA .ASPGA001     DELETED          DDNAME=FT06F001            9 EXCPS
IEF285I   VOL SER NOS= ASP760.
IEF285I   SYS81008.T171125.RV001. ZBNAMAAA .DISPLAY      DELETED          DDNAME=FT07F001            3 EXCPS
IEF285I   VOL SER NOS= SCR001.
IEF285I   SYS81008.T171125.RV001. ZBNAMAAA .ASPGA002     DELETED          DDNAME=FT09F001           20 EXCPS
IEF285I   VOL SER NOS= ASP761.
IEF285I   GJMAS.DYNAMICS.DATA                            KEPT             DDNAME=FT10F001           18 EXCPS
IEF285I   VOL SER NOS= DISK09.
IEF285I   SYS81008.T171125.RV001. ZBNAMAAA .DYN          DELETED          DDNAME=FT11F001            0 EXCPS
IEF285I   VOL SER NOS= SCR001.
IEF285I   ORBIT.GTDS.SLP1920.DATA                        KEPT             DDNAME=FT14F001            3 EXCPS
IEF285I   VOL SER NOS= DISK00.
IEF285I   ORBIT.GTDS.JACCHIA.DATA                        KEPT             DDNAME=FT15F001            0 EXCPS
IEF285I   VOL SER NOS= DISK19.
IEF285I   GJMAS.ERRMSG.DATA                              KEPT             DDNAME=FT17F001            0 EXCPS
IEF285I   VOL SER NOS= DISK02.
IEF285I   SYS81008.T171125.RV001. ZBNAMAAA .CURR         DELETED          DDNAME=FT18F001            1 EXCPS
IEF285I   VOL SER NOS= SCR001.
IEF285I   GJMAS.NEWS.DATA                                KEPT             DDNAME=FT19F001            2 EXCPS
IEF285I   VOL SER NOS= DISK09.
IEF285I   ORBIT.GTDS.EARTHFLD.DATA                       KEPT             DDNAME=FT20F001            0 EXCPS
IEF285I   VOL SER NOS= DISK19.
IEF285I   ORBIT.GTDS.LUNARFLD.DATA                       KEPT             DDNAME=FT21F001            0 EXCPS
IEF285I   VOL SER NOS= DISK19.
IEF285I   ORBIT.GTDS.ATMOSDEN.DATA                       KEPT             DDNAME=FT22F001            0 EXCPS
IEF285I   VOL SER NOS= DISK19.
IEF285I   SYS81008.T171125.RV001. ZBNAMAAA .A0000333     DELETED          DDNAME=FT26F001            0 EXCPS
IEF285I   VOL SER NOS= SCR001.
IEF285I   GJMAS.STATION.DATA                             KEPT             DDNAME=FT29F001            0 EXCPS
IEF285I   VOL SER NOS= DISK17.
IEF285I   ORBIT.GTDS.D24HOUR.DATA                        KEPT             DDNAME=FT30F001            0 EXCPS
IEF285I   VOL SER NOS= DISK19.
IEF285I   ORBIT.GTDS.TIMCOF.DATA                         KEPT             DDNAME=FT38F001            4 EXCPS
IEF285I   VOL SER NOS= DISK00.
IEF285I   GJMAS.MONRES.DATA                              KEPT             DDNAME=FT49F001            4 EXCPS
IEF285I   VOL SER NOS= DISK09.
IEF285I   ATTIT.GESSMSG.DATA                             KEPT             DDNAME=GESSMSG -1          4 EXCPS
IEF285I   VOL SER NOS= DISK52.
IEF285I   GJMAS.GMASMSG.DATA                             KEPT             DDNAME=GESSMSG -2          0 EXCPS
IEF285I   VOL SER NOS= DISK02.
$$$
$$$ CHARGE TIME          CPU=0000.070 MIN  I/O=0000.180 MIN
$$$ I/O TIME IN SEC.     DRUM=0000.88 DISK=0000.50 CELL=0000.00 TAPE=0000.00 GRAF=0000.00  ASP=0000.49
$$$ I/O COUNTS           TOTAL-EXCPS=0000325    MISC-DEVICES=000000  CARDS=00000
$$$ REGION               START=1536K  SIZE=0376K  MAX-USED=0344K  PERCENT-USED=091
$$$ ZBKAPAAA/GO          STEP=001  PGM=GMAS                  START=81-08-01/17.35.33  STOP=81-08-01/17.36.23
$$$
$$$
$$$
$$$ CHARGE TIME          CPU=0000.070 MIN  I/O=0000.180 MIN
$$$ I/O TIME IN SEC.     DRUM=0000.88 DISK=0000.50 CELL=0000.00 TAPE=0000.00 GRAF=0000.00  ASP=0000.49
$$$ I/O COUNTS           TOTAL-EXCPS=0000325    MISC-DEVICES=000000
$$$ ZBNAMAAA             ROOM=J  CLASS=A  PRTY=08            START=81-08-01/17.35.33  STOP=81-08-01/17.36.23
$$$
$$$ NASA/GSFC 360/95 61      SYSTEM=MVT-21.8     FOR PROBLEM ASSISTANCE, CALL PAC ON 344-6768
AMO309 JOB 2648 ( ZBNAMAAA ) IN BREAKDOWN
```

Figure 2-4. Messages From the Computer (3 of 3)

```
                                    USER CARD INPUT FOR THIS RUN

 &GMASEX  SEQNAM='CARDS',IBATCH=1,&END
CRIVE1
PRFCCN
CRBINP
 &ORBIN
  IELEM=1,
  ICORD=2,
  ELEM=800729.,70958.4,
       7000.,200.,300.,
       0.,3.3,8.0,
  PROPM='CCWELL',
  H=100.,
  ISTOP=1,
  STPVAL=86400.,
 &END
CRBIT
EOF
```

Figure 2-5. First Page of GMAS Output

```
**NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS*

    *************GMAS STATUS AS OF 12/09/80 *******************            00000010
    * * * * * * *N E W S   N E W S   N E W S   N E W S* * * * * * *       00000020
    * * * *ON MONDAY DECEMBER 15, 1980 A NEW OVERLAY FOR * * * *          00000030
    * * * *AVGVOP PROPAGATOR MODULE WILL BE ADOPTED. USER * * * *         00000040
    * * * *SHOULD SEE SIGNIFICANT DECREASE IN I/O TIME.  * * * *          00000050
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *         00000060
    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *         00000070

**NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS****NEWS*
```

Figure 2-6. Second Page of GMAS Output

```
EXECUTING(MERGED) AUTOMATIC SEQUENCE
     DRIVE1
     PRFCON
     ORBINP
      $ORBIN
          ISTOP=1,17*0,
          STPVAL=86400.D0,17*0.D0,
          IREPT=18*0,
          ISKIP=18*0,
          MULTI=18*0,
          ITERM=5*0,
          PARMM='GPARM',    NOUT=2,0,0,      IPARMS=1,9*0,
          IPMOPT=0,
          PROPM='COWELL',       IPOPT=1,       IDEL=2,
          IPRNT=0,
          IELEM=1,  ICORD=2,  ICENT=1,
          ELEM=750729.D0,       70958.4D0,
          6526.439D0,      0.D0,            0.D0,
          0.D0,            3.24286D0,       8.02636BD0,
          M=0.D0,          IDYN=1,
          SARA=270.0D0,        SDEC=66.554241 3D0,   IACORD=1,

         IELEM=1,
         ICORD=2,
         ELEM=800729.,70958.4,
              7000.,200.,300.,
              0.,3.3,8.0,
         PROPM='COWELL',
         M=100.,
         ISTOP=1,
         STPVAL=86400.,
         &END
        ORBIT
        EOF
&COWLINP
STATE1=  2444449.79881289788     ,  1.0000000000000000     ,  7002.05168498762214    ,  152.449469457884991    ,
  279.340938686965728    ,  .460211629932524021D-01,  3.29956107042622948    ,  8.00004867454711754    ,  1.00000000000000000    ,
  100.000000000000000    ,STPVAL=  86400.0000000000000    ,  .0                    ,  .0                    ,
  .0                     ,  .0                     ,  .0                    ,  .0                    ,  .0                    ,
  .0                     ,  .0                     ,  .0                    ,  .0                    ,  .0                    ,
  .0                     ,  .0                     ,  .0                    ,  .0                    ,  .0                    ,
&END
```

Figure 2-7. GMAS Merged Automatic Sequence

```
      ****** DEFINITION OF ABBREVIATIONS ******
      AND = ASCENDING NODE DOT IN DEG/DAY                                  (33)
      AP  = ARGUMENT OF PERIFOCUS IN DEG                                   ( 5)
      APD = ARGUMENT OF PERIFOCAL DOT IN DEG/DAY                           (32)
      APH = OSCULATING APOFOCAL GEODETIC HEIGHT IN KM                      (22)
      APR = APOFOCAL RADIUS IN KM                                          (27)
      AZ  = VELOCITY AZIMUTH IN DEG                                        (16)
      CJ  = ENERGY IN KM**2/SEC**2                                         (28)
      DCE = DECLINATION OF VEHICLE WITH RESPECT TO EARTH
            IN DEG                                                         (55)
      DCM = DECLINATION OF VEHICLE WITH RESPECT
            TO MOON IN DEG                                               (52)
      DCS = DECLINATION OF VEHICLE WITH RESPECT TO SUN IN DEG              (49)
      DEC = DECLINATION IN DEG                                             (14)
DX.DY.DZ = VELOCITY IN KM/SEC                                      (10,11,12)
      EA  = ECCENTRIC ANOMALY IN DEG                                       (19)
      ECC = ECCENTRICITY                                                   ( 2)
      EMS = EARTH-MOON-SUN ANGLE IN DEG                                    (42)
      EST = EARTH SUBTENDED ANGLE FROM VEHICLE IN DEG                      (43)
      EVM = EARTH-VEHICLE-MOON ANGLE IN DEG                                (41)
      EVS = EARTH-VEHICLE-SUN ANGLE IN DEG                                 (40)
      HGT = GEODETIC HEIGHT IN KM                                          (29)
      INC = INCLINATION IN DEG                                             ( 3)
      LAT = LATITUDE IN DEG                                                (23)
      LON = LONGITUDE IN DEG                                               (24)
      MA  = MEAN ANOMALY IN DEG                                            (20)
      MM  = MEAN MOTION IN DEG/MIN                                         (31)
      MST = MOON SUBTENDED ANGLE FROM VEHICLE IN DEG                       (45)
      P   = PERIOD IN MIN                                                  (30)
      PH  = OSCULATING PERIFOCAL GEODETIC HEIGHT IN KM                     (21)
      PR  = PERIFOCAL RADIUS IN KM                                         (26)
      R   = MAGNITUDE OF POSITION VECTOR IN KM                             (17)
      RA  = RIGHT ASCENSION IN DEG                                         (13)
      RAE = RIGHT ASCENSION OF VEHICLE WITH RESPECT TO EARTH
            IN DEG                                                         (54)
      RAM = RIGHT ASCENSION OF VEHICLE WITH RESPECT TO MOON IN
            DEG                                                          (51)
      RAN = RIGHT ASCENSION OF ASCENDING NODE IN DEG                       ( 4)
      RAS = RIGHT ASCENSION OF VEHICLE WITH RESPECT TO SUN IN
            DEG                                                          (48)
      RE  = MAGNITUDE OF POSITION VECTOR OF VEHICLE
            WITH RESPECT TO EARTH IN KM                                  (53)
      RM  = MAGNITUDE OF POSITION VECTOR OF VEHICLE WITH RESPECT
            TO MOON IN KM                                                (50)
      RS  = MAGNITUDE OF POSITION VECTOR OF VEHICLE WITH RESPECT
            TO SUN IN KM                                                 (47)
      SAE = SPIN AXIS-EARTH ANGLE IN DEG                                   (37)
      SAM = SPIN AXIS-MOON ANGLE IN DEG                                    (39)
      SAS = SPIN AXIS-SUN ANGLE IN DEG                                     (38)
      SLR = SEMI-LATUS RECTUM IN KM                                        (25)
      SMA = SEMI-MAJOR AXIS IN KM                                          ( 1)
      SST = SUN SUBTENDED ANGLE FROM VEHICLE IN DEG                        (44)
      SVN = SUN-VEHICLE-ORBIT NORMAL ANGLE IN DEG
            (SUN ANGLE)                                                  (46)
      TA  = TRUE ANOMALY IN DEG                                            ( 6)
      TSP = TIME SINCE PERIGEE IN MIN                                      (36)
      V   = MAGNITUDE OF VELOCITY VECTOR IN KM/SEC                         (18)
      VAP = VELOCITY AT APOFOCUS IN KM/SEC                                 (34)
      VP  = VELOCITY AT PERIFOCUS IN KM/SEC                                (35)
      VPA = VERTICAL FLIGHT PATH ANGLE IN DEG                              (15)
    X.Y.Z = POSITION IN KM                                           (7, 8, 9)
            *** TIME IS IN UTC TIME ***

            ( ) DENOTE IPARMS REFERENCE NUMBERS
```

Figure 2-8. List of GMAS Abbreviations

DATE= 800729 7 9 58.400 (JULIAN=2444449.79859) TIME FROM EPOCH= 0 DAYS 0 HRS 0 MIN 0.000SEC TIME
CENTRAL BODY= EARTH COORDINATE SYSTEM= TRUE EQ DATE LOCAL TIME= 3 HRS 9 MIN 58.400 SEC

SMA	10261.27862	ECC	.3205109338	INC	67.58505070	RAN	.6240900582	AP	350.7062449	TA	11.94740707
X	7000.000000	Y	200.0000000	Z	300.0000000	DX	.1231488869D-12	DY	3.300000000	DZ	8.000000000
RA	1.636577042	DEC	2.453031865	VPA	87.10836721	AZ	22.43662722	R	7009.279564	V	8.653900855
EA	8.585347521	MA	5.843935436	PH	594.2866285	APH	7171.990615	LAT	2.468048080	LON	307.0512712
SLR	9207.165598	PR	6972.426628	APR	13550.13061	C3	-38.84512396	HGT	631.1789123	P	172.4097580
MM	2.088048868	APD	-.3198608008	AND	-.8935043639	VAP	4.470833449	VP	8.688564314	TSP	2.798754150

DATE= 800730 7 9 58.400 (JULIAN=2444450.79859) TIME FROM EPOCH= 1 DAYS 0 HRS 0 MIN 0.000SEC TIME
CENTRAL BODY= EARTH COORDINATE SYSTEM= TRUE EQ DATE LOCAL TIME= 14 HRS 9 MIN 58.400 SEC

SMA	10248.44279	ECC	.3192188514	INC	67.57589838	RAN	359.7165435	AP	350.3714830	TA	156.8676008
X	-10947.43440	Y	2744.711788	Z	6519.901891	DX	-3.216246548	DY	-1.303656781	DZ	-5.197655400
RA	165.9253794	DEC	30.01436774	VPA	79.90316339	AZ	153.8619787	R	13034.14298	V	4.718979012
EA	148.1728992	MA	138.5064320	PH	591.6327412	APH	7148.972832	LAT	30.09586773	LON	110.3541429
SLR	9199.534431	PR	6969.772741	APR	13527.11283	C3	-38.89377619	HGT	6661.360366	P	172.0863581
MM	2.091972914	APD	-.3203316301	AND	-.8970165102	VAP	4.476584719	VP	8.688269882	TSP	66.20852070

Figure 2-9. GMAS Parameter Report

SIZE=0376K indicates that the size of the region allocated by the computer to run this job was 376K bytes of core. This was input via the // EXEC GMAS,REGION.GO=376K card (see Section 2.2.1).[1] MAX-USED=344K indicates the actual region size used to run the job. If a difference of more than 50K bytes exists between these numbers, the computer will refuse to continue processing the job.

Figure 2-5 shows the first page of GMAS output. This page lists the user input for the run.

Figure 2-6 shows the next page of GMAS output, which is the GMAS news page. This page contains material that is intended to be informative to the user.

Figure 2-7 shows the merged automatic sequence. The variables indented under &ORBIN are the default orbit input variables, followed by the user's input. Printed below the merged automatic sequence is some information from the orbit input processor. The information on this page can be of use to the experienced GMAS user, although it may seem confusing to the novice.

Figure 2-8 lists the abbreviations for the standard GMAS output parameters.

Figure 2-9 shows the GMAS parameter report. Although all other printout pages are informative, the primary purpose of the run is this parameter report. The parameter information generated for each step is separated by asterisks. The first section specifies the values of various parameters at epoch. (Figure 2-8 lists the parameter abbreviations and units.) The second and final section specifies the values of these parameters 1 day after epoch. Each section contains the time from epoch and the local time.

---

[1]Core is allocated in even-numbered increments. Thus, REGION.GO=375K results in an allocation of 376K bytes.

The parameter report is followed by a number of Graphic Executive Support System (GESS) output displays, which are discussed in Section 4 of the User's Guide. These output displays are primarily used for error detection.

SECTION 3 - USING GMAS TO PROPAGATE A SATELLITE ORBIT

Section 2 of this primer shows how GMAS can be used to propagate a specific satellite orbit for 1 day (Problem 1). In propagating this orbit, the user specifies the values of certain variables (e.g., the satellite's initial orbital parameters, epoch, length of propagation) through card input while using the internally defaulted values of other variables. Many more orbit propagation input options are available through GMAS than are shown in the Problem 1 example. This section discusses the orbit propagation input options available to the user, which can be classified into two categories: (1) propagation control input, discussed in Section 3.1, and (2) force model input, discussed in Section 3.2.

3.1 PROPAGATION CONTROL INPUT

Propagation control input is accomplished by means of the variable cards entered between the &ORBIN card and the &END card in the input deck (see Figure 2-1). This input can be classified into five categories: input state, propagator options, stopping conditions, output options, and general flags. Table C-1 of the User's Guide lists and defines all of the available propagation control input variables in their respective categories and specifies their default values. The following sections discuss the use of these variables, which are presented by category. It is to be noted that commas must immediately follow variable values.

3.1.1 INPUT STATE

3.1.1.1 Variable Description

Variables in the input state category include ELEM, IELEM, ICORD, ICENT, IACORD, SARA, and SDEC.

ELEM is an eight-element array. The first position contains the packed year, month, and day (YYMMDD.) of epoch. The second position contains the packed hour, minute, and second (HHMMSS.S) of epoch. The positions allowed for each quantity (day, year, and so forth) must be zero filled, if necessary. For example, if the satellite epoch is 5:13 a.m. on January 27, 1981, the first two positions in the ELEM array are ELEM=810127.,051300.,. (It should be noted that all numbers in the ELEM array must contain decimal points.) The next six positions in the ELEM array contain the state of the satellite at epoch. Since there are various ways of describing the state of the satellite, some clarifications must be made using variables IELEM, ICORD, and ICENT. Table C-1a of the User's Guide specifies the values of these variables.

IELEM specifies the type of the initial state elements. If the six initial elements are Cartesian elements (x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$), the user sets IELEM=1. Since 1 is the default value for IELEM, the user need not include IELEM in his list of variables to be entered between the &ORBIN card and the &END card in the input deck. If the six initial elements are Keplerian elements with mean anomaly (a, e, i, Ω, ω, M), the user sets IELEM=2. Table C-2 of the User's Guide should be used to determine the ELEM array for a given IELEM value.

ICORD specifies the coordinate system to which the initial state elements are referenced. Satellite elements are usually given in the true Earth equator and equinox of date coordinate system. If this is the case, the user sets ICORD=2. Since 2 is the default value for ICORD, the user need not include this variable in his list. If the initial state elements are given in the mean Earth equator and equinox of 1950.0 system, the user sets ICORD=1.

ICENT identifies the center of the input element coordinate system. The Earth is usually the center (origin) of the coordinate system to which the elements are referenced. If this is the case, the user sets ICENT=1 (or omits ICENT from his list, since 1 is the default value for this variable.) If the initial element coordinates are centered at Mars, the user sets ICENT=4. It should be noted that variable ICENT refers only to the center of the input coordinate system and not to the propagation central body. The latter is identified by variable IBODY, which is set in the force model input section (see Section 3.2).

IACORD, SARA, AND SDEC identify initial spacecraft attitude conditions. Table C-1a of the User's Guide specifies the input values for these variables. If the user is concerned only with the satellite orbit, these variables can be ignored.

3.1.1.2 Example of Variable Use

A brief example of the use of variables ELEM, IELEM, ICORD, and ICENT is presented in the following case, in which the objective is to propagate a satellite orbit, given the following:

Epoch:	January 27, 1981; 05:20:00 GMT
Coordinate system of input:	Earth-centered mean ecliptic and equinox of 1950.0
Elements:	Keplerian: a = 8000 kilometers e = 0.1 i = 10 degrees Ω = 20 degrees ω = 30 degrees M = 0 degrees

To enter the necessary information correctly, the user must use the following variables and values:

```
ICORD=3,ICENT=1,IELEM=2,
ELEM=810127.,052000.,
```

```
8000.,.1,10.,
20.,30.,0.0,
```

The following points should be noted:

- IELEM, ICORD, and ICENT are integer variables and do not have decimal points, whereas ELEM values do.
- The variables in the list following the &ORBIN card are always followed by commas (even at the end of a line).
- No line can extend beyond column 64 of the card.

3.1.2 PROPAGATOR OPTIONS

3.1.2.1 Variable Description

Variables in the propagator options category include PROPM, IPOPT, and H.

PROPM identifies the propagator to be used. The propagator name must be in quotes (e.g., PROPM='AVGVOP'). If variable PROPM is omitted from the user's input list, the Cowell propagator is used by default.

IPOPT is used in conjunction with the analytic propagator or for special element conversion. The use of this variable is discussed in subsequent sections.

H specifies the step size (in seconds) of the propagator to be used. For the time-regularized Cowell propagator, H represents the number of steps per revolution.

3.1.2.2 Example of Variable Use

To propagate an orbit using the time-regularized Cowell propagator with 200 steps per revolution, the user must use the following variables and values:

```
PROPM='TRCOWL',
H=200.,
```

3.1.3 STOPPING CONDITIONS

3.1.3.1 Introduction

When the propagator achieves a stopping condition, it prints out the satellite information specified by variable NOUT (see Section 3.1.4). Problem 1 (Section 2.1) includes only one stopping condition (i.e., 1 day). Therefore, the printed output from Problem 1 includes only the initial output parameter report (which is always included) and the output parameter report after 1 day of propagation (see Figure 2-9). By setting stopping condition variables properly, it is possible to retrieve satellite state information at many selected points during a propagation. Table C-1c of the User's Guide describes the stopping condition variables.

Before proceeding in this discussion of stopping conditions, the distinction between the terms "stop" and "terminal stop" must be noted. A "stop" on a condition means that an output parameter report is generated and the propagator continues to the next stop. (An output parameter report (see Figure 2-9) is generated for each stop.) A "terminal stop" means that orbit propagation terminates after the parameter report is generated.

3.1.3.2 Variable Description and Examples of Use

Variables in the stopping conditions category include ISTOP, STPVAL, ISKIP, IREPT, MULTI, and ITERM.

ISTOP and STPVAL are 18-element arrays that describe up to 18 stopping conditions. Table C-3 of the User's Guide specifies the available values of these variables. ISTOP contains the stopping conditions, and STPVAL contains the corresponding values under which these conditions are met. For example, to stop when the distance of the satellite from the Earth is 500 kilometers and increasing, the user sets ISTOP=1, STPVAL=500.,.

A more complex example of the use of ISTOP and STPVAL is the following case, in which the objective is to stop a satellite at each of the following points:

- Two hours (7200 seconds) after epoch
- When the satellite's distance from Earth is 6000 kilometers and increasing
- First ascending node
- Point of farthest approach from the central body
- When the osculating argument of perigee is 10 degrees

To accomplish this, the user must use the following variables and values:

```
ISTOP=1,3,11,14,27,
STPVAL=7200.,6000.,0.,0.,10.,
```

The following points should be noted:

- STPVAL values must include decimal points.
- Values are separated by commas and end with a comma.
- Stopping condition 11 does not need a value, so 0. is used to fill the corresponding STPVAL position.
- The user does not need to fill in all the positions of the 18-element ISTOP and STPVAL arrays. The positions not included as input are defaulted to 0.

It should also be noted that values in the ISTOP array cannot have embedded zeros. For example ISTOP=1,2,0,3,4 causes the system to ignore the underlined stopping conditions and has the same effect as ISTOP=1,2.

ISKIP, IREPT, and MULTI are 18-element arrays that contribute to the establishment of the desired stopping configuration. The best explanation of the use of these variables is

an example such as the following case, in which the variables and values used are

```
ISTOP=1,
STPVAL=86400.,
ISKIP=3,
IREPT=5,
MULTI=2,
```

If these variables and values are added to the user's list, the results are as follows: GMAS stops on time (ISTOP=1) every day (STPVAL=86400.); the output parameter report is skipped three times (ISKIP=3) and then printed five times (IREPT=5), and this sequence (three skips, five prints) is repeated two times (MULTI=2). Figure 3-1 illustrates this configuration using a timeline to show the points at which output is generated.

The following points should be noted:

- An initial output parameter report is always generated at epoch (0).
- Propagation terminates at 16 days when all conditions have been met.

The foregoing example uses only one element of the ISTOP array. Stopping configurations are more interesting when they involve multiple stopping conditions, each with its own set of ISKIP, IREPT, and MULTI values. To illustrate the use of these variables in a multiple stopping condition environment, the following case is considered, in which the variables and values used are

```
ISTOP=1,1,1,
STPVAL=100.,175.,1100.,
ISKIP=2,1,0,
IREPT=3,1,3,
MULTI=4,3,0,
```

The relationship of these variables and values is easier to understand when they are presented in tabular format, as in

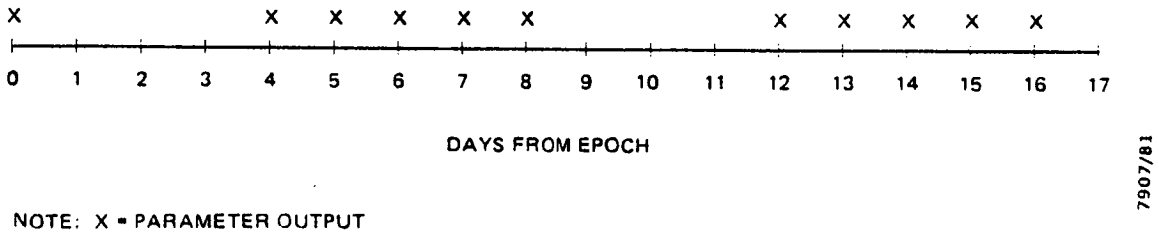


Figure 3-1. Stopping Configuration

Table 3-1. In this table, the same variable values presented above are more readily seen as sets of stopping conditions, labeled A, B, and C. In Figure 3-2, corresponding labels A, B, and C illustrate the effect of these sets of stopping conditions in a timeline that shows when stopping occurs.

ITERM is an five-element array that specifies the terminal stop of propagation. If ITERM=N, the Nth stopping condition is designated to be terminal. This means that when ISKIP, IREPT, and MULTI have been satisfied for the Nth condition, the propagation terminates regardless of the completion of other conditions. If, for the preceding example, the user sets ITERM=1, propagation terminates at 2000 seconds (the last A in Figure 3-2). If the user sets ITERM=2, propagation terminates at 1050 seconds (the last B in Figure 3-2). If the user sets ITERM=3, propagation terminates at the same time as in the original example (3300 seconds).

ITERM can contain up to five values. If ITERM=1,3,5, propagation terminates on the first stopping condition in this set that is met. As an example, the following case is considered:

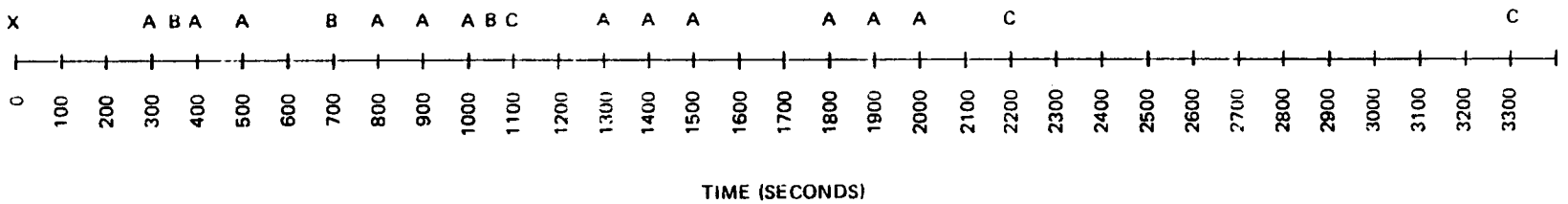
```
ISTOP=1,1,1,1,1,
STPVAL=5000.,4000.,2000.,500.,10000.,
ITERM=1,3,5,
```

In this case the propagator stops at 500 seconds and at 2000 seconds. Since 2000 seconds (the third stopping condition) is one of the designated terminal stopping conditions, propagation terminates at this point. Therefore, the stop values of 4,000, 5,000, and 10,000 seconds do not occur.

Table 3-1. Sample Stopping Logic

VARIABLE	A	B	C
ISTOP	1	1	1
STPVAL	100	175	1100
ISKIP	2	1	0
IREPT	3	1	3
MULTI	4	3	0

7907/81



7097/81

NOTE:

A, B, C = STOPPING CONDITION SET LABELS (SEE TABLE 3-1) INDICATING STOPS

X = INITIAL INPUT

Figure 3-2. Multiple Stopping Condition Configuration

It should be noted that the propagator continues to propagate until one of the following three conditions is met:

1. All stopping conditions have been satisfied.
2. A terminal stopping condition has been satisfied.
3. One of the two default terminal conditions (i.e., impact on the propagation central body, 3-year time limit) has been met.

The 3-year time limit and impact constraints can be changed by including variables TOF1 and RAD1 under the &DYNSEC card (see Table D-2 of the User's Guide). The procedure for making these changes is discussed in Section 3.2 of this document.

3.1.4 OUTPUT OPTIONS

Variables in the output options category include PARMM, NOUT, and IPARMS. Table C-1d of the User's Guide describes these variables.

PARMM specifies the alphanumeric name of the parametric calculation module. For normal orbit propagation output, the user sets PARMM='GPARM'. Since this is the default value for PARMM, the user need not include PARMM in his input list. Other values of PARMM are for special, unconventional types of output, which are discussed in subsequent sections relative to specific cases.

NOUT is a three-element array that determines the type of output parameter report occurring at each stop. NOUT(2), which specifies the output coordinate system reference, defaults to the input coordinate system. NOUT(3), which specifies the output central body reference, defaults to the integrator central body (usually, Earth). In general, the user can ignore the second and third elements of NOUT, leaving them to their default values.

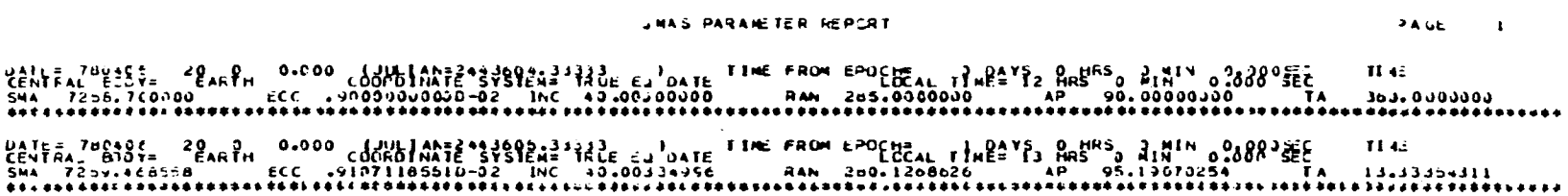
The first position in the NOUT array determines the type and level of the output report to be generated. Table C-6a of the User's Guide describes the various standard levels of output available through GMAS. Figure 2-8 of this primer contains a list of the parameter abbreviations used in the output. Figures 3-3 through 3-6 show the various levels of output reports generated for the propagation of a satellite for 1 day. If NOUT=-1, the output is the same as the level 1 output (NOUT=1), except that it also contains the Brouwer mean elements. If the user does not include NOUT in his input list, the output report generated will be level 2 (NOUT=2), since 2 is the default value for NOUT.

IPARMS, which is used only when NOUT=4, 5, or 6, allows the user to list up to 10 parameters to be generated. If the user is not satisfied with the printed output obtainable by setting NOUT=1, NOUT=2, or NOUT=3, he can set NOUT=4 and choose up to 10 parameters to be printed. The procedure for this is as follows:

1. Set NOUT=4.
2. Choose (from Table C-6c of the User's Guide) up to 10 parameters to be printed at each stop.
3. Set IPARMS=N_1,N_2,N_3, and so forth, where the letters N represent the numbers corresponding to the parameters selected from Table C-6c.

For example, if the user desires to have printed out (1) the magnitude of the spacecraft position vector, (2) the magnitude of the spacecraft velocity vector, (3) the energy, and (4) the orbital period, he sets NOUT=4, IPARMS=17,18,28,30. Figure 3-7 shows the resulting output for a 1-day propagation. Since the user does not include values for NOUT(2) and NOUT(3), their defaulted values result.

The remaining two values of NOUT (5 and 6) are used in special cases when the user desires that information be passed

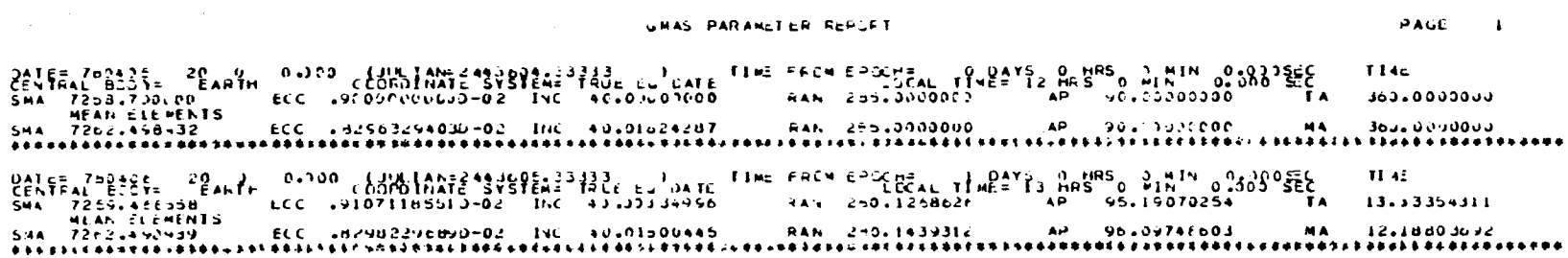


```
                                    GMAS PARAMETER REPORT                                          PAGE     1

DATE= 780426   20  0   0.000   (JULIAN=2443604.33333    )   TIME FROM EPOCH=   3 DAYS  0 HRS  3 MIN  0.000 SEC      TIME
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE          LOCAL TIME= 12 HRS  0 MIN  0.000 SEC
SMA   7256.700000       ECC  .9000000000D-02   INC  40.00000000       RAN  285.0000000       AP   90.00000000       TA   360.0000000
*********************************************************************************************************************************

DATE= 780426   20  3   0.000   (JULIAN=2443607.33333    )   TIME FROM EPOCH=   0 DAYS  0 HRS  3 MIN  0.000 SEC      TIME
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE          LOCAL TIME= 15 HRS  0 MIN  0.000 SEC
SMA   7259.468568       ECC  .9107118551D-02   INC  40.00334956       RAN  280.1268626       AP   95.19070254       TA   13.33354311
*********************************************************************************************************************************
```

Figure 3-3. Level 1 Output (NOUT=1)



```
                                    GMAS PARAMETER REPORT                                          PAGE     1

DATE= 780426   20  0   0.000   (JULIAN=2443604.33333    )   TIME FROM EPOCH=   3 DAYS  0 HRS  3 MIN  0.000 SEC      TIME
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE          LOCAL TIME= 12 HRS  0 MIN  0.000 SEC
SMA   7258.700000       ECC  .9000000000D-02   INC  40.00000000       RAN  285.0000000       AP   90.00000000       TA   360.0000000
     MEAN ELEMENTS
SMA   7262.468432       ECC  .8256329403D-02   INC  40.01624287       RAN  285.0000000       AP   90.00000000       MA   360.0000000
*********************************************************************************************************************************

DATE= 780426   20  3   0.000   (JULIAN=2443607.33333    )   TIME FROM EPOCH=   0 DAYS  0 HRS  3 MIN  0.000 SEC      TIME
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE          LOCAL TIME= 15 HRS  0 MIN  0.000 SEC
SMA   7259.468568       ECC  .9107118551D-02   INC  40.00334956       RAN  280.1268626       AP   95.19070254       TA   13.33354311
     MEAN ELEMENTS
SMA   7262.490439       ECC  .8249822476890-02   INC  40.01500445       RAN  280.1439312       AP   96.09746603       MA   12.18803692
*********************************************************************************************************************************
```

Figure 3-4. Level 1 Output With Brouwer Mean Elements (NOUT=-1)

```
                                   SMAS PARAMETER REPORT                                                        PAGE    1

DATE= 780405    20  0    0.000  (JULIAN=2443604.33333    )    TIME FROM EPOCH=    0 DAYS  0 HRS  0 MIN  0.000SEC      TIME
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE           LOCAL TIME= 12 HRS  0 MIN   0.000 SEC
SMA   7258.700000     ECC  .9000000000D-02  INC   40.00000000      RAN  265.0000000     AP   90.00000000     TA    360.0000000
X     5322.678646     Y    1426.207445      Z     4623.810201      DX  -1.935283342     DY   7.222575761     DZ   -.4609854165D-12
RA    15.00000000     DEC  40.00000000      VPA   90.00000000      AZ   90.00000000     R    7193.371700     V     7.477360647
EA    360.0000000     MA   360.0000000      PH    815.2317000      APH  945.8883000     LAT  40.16787029     LON   241.2351049
SLR   7258.112045     FF   7193.371700      APR   7324.028000      C3  -54.91330242     HGT  824.0896539     P     102.5765040
MM    3.505575643     AFD  6.125947167      AND  -4.854959155      VAP  7.343968883     VP   7.477360647     TSP   102.5765040
**********************************************************************************************************************************

DATE= 780406    20  0    0.000  (JULIAN=2443605.33333    )    TIME FROM EPOCH=    1 DAYS  0 HRS  0 MIN  0.000SEC      TIME
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE           LOCAL TIME= 13 HRS  0 MIN   0.000 SEC
SMA   7259.466556     ECC  .9107118551D-02  INC   40.00334956      RAN  250.1266625     AP   95.19070254     TA    13.33354311
X     4742.559368     Y    3169.184142      Z     4385.610601      DX  -3.027156647     DY   6.665134314     DZ   -1.517338560
RA    33.75234319     DEC  37.55529535      VPA   89.88072009      AZ   134.5248450     R    7195.100112     V     7.475962895
EA    13.21373442     MA   13.09445929      PH    815.2157174      APH  947.4413939     LAT  37.72005075     LON   259.0018489
SLR   7258.666461     FF   7193.355717      APR   7325.581399      C3  -54.50766874     HGT  824.5325293     P     102.5927958
MM    3.505018321     AFD  6.125828188      AND  -4.852581121      VAP  7.342753264     VP   7.477765653     TSP   2.731658854
**********************************************************************************************************************************
```

Figure 3-5. Level 2 Output (NOUT=2)

```
                                   SMAS PARAMETER REPORT                                                        PAGE    1

DATE= 780405    20  0    0.000  (JULIAN=2443604.33333    )    TIME FROM EPOCH=    0 DAYS  0 HRS  0 MIN  0.000SEC      TIME
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE           LOCAL TIME= 12 HRS  0 MIN   0.000 SEC
SMA   7258.700000     ECC  .9000000000D-02  INC   40.00000000      RAN  265.0000000     AP   90.00000000     TA    360.0000000
X     5322.678646     Y    1426.207445      Z     4623.810201      DX  -1.935283342     DY   7.222575761     DZ   -.4609854165D-12
RA    15.00000000     DEC  40.00000000      VPA   90.00000000      AZ   90.00000000     R    7193.371700     V     7.477360647
EA    360.0000000     MA   360.0000000      PH    815.2317000      APH  945.8883000     LAT  40.16787029     LON   241.2351049
SLR   7258.112045     FF   7193.371700      APR   7324.028000      C3  -54.91330242     HGT  824.0896539     P     102.5765040
MM    3.505575643     AFD  6.125947167      AND  -4.854959155      VAP  7.343968883     VP   7.477360647     TSP   102.5765040
SAL   120.7201145     SAS  90.00201608      SAN   89.39340502      EVS  146.1645007     EVM  133.1634007     EMS   157.5557194
LST   124.5152385     SST  .5327927964      MST   .5316588350      SVN  103.6292187
RS    149652515.1     RAS  134.4387636      DCS  -6.165606027      RM   372153.1424     RAM  173.0538536     DCM   2.324170869
RE    7193.371700     RAE  15.00000000      DCE   40.00000000
**********************************************************************************************************************************

DATE= 780406    20  0    0.000  (JULIAN=2443605.33333    )    TIME FROM EPOCH=    1 DAYS  0 HRS  0 MIN  0.000SEC      TIME
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE           LOCAL TIME= 13 HRS  0 MIN   0.000 SEC
SMA   7259.466556     ECC  .9107118551D-02  INC   40.00334956      RAN  250.1266625     AP   95.19070254     TA    13.33354311
X     4742.559368     Y    3169.184142      Z     4385.610601      DX  -3.027156647     DY   6.665134314     DZ   -1.517338560
RA    33.75234319     DEC  37.55529535      VPA   89.88072009      AZ   134.5248450     R    7195.100112     V     7.475962895
EA    13.21373442     MA   13.09445929      PH    815.2157174      APH  947.4413939     LAT  37.72005075     LON   259.0018489
SLR   7258.666461     FF   7193.355717      APR   7325.581399      C3  -54.50766874     HGT  824.5325293     P     102.5927958
MM    3.505018321     AFD  6.125828188      AND  -4.852581121      VAP  7.342753264     VP   7.477765653     TSP   2.731658854
SAL   112.5600522     SAS  90.00236551      SAN   [illegible]      EVS  144.7665513     EVM  130.1620066     EMS   170.1142577
LST   124.6622546     SST  .5026389033      MST   .5203523025      SVN  123.2741613
RS    149735768.5     RAS  135.3523870      DCS  -6.347103690      RM   373254.2513     RAM  185.8743496     DCM  -1.993752123
RE    7195.100112     RAE  33.75234319      DCE   37.55529535
**********************************************************************************************************************************
```

Figure 3-6. Level 3 Output (NOUT=3)

```
                                            GPAS PARAMETER REPORT                                                  PAGE    1

DATE= 750729     7  9  58.400  (JULIAN=2442622.75859     )      TIME FROM EPOCH=     0 DAYS  0 HRS  0 MIN  0.000SEC     TIME
CENTRAL BODY=    EARTH          COORDINATE SYSTEM= TRUE EQ DATE                   LOCAL TIME=  3 HRS  9 MIN  58.400 SEC
   R  6526.439000          V  8.656715558          C3 -43.21077672          P  126.6781664
*****************************************************************************************************************************

DATE= 750730     7  9  58.400  (JULIAN=2442623.75859     )      TIME FROM EPOCH=     1 DAYS  0 HRS  0 MIN  0.000SEC     TIME
CENTRAL BODY=    EARTH          COORDINATE SYSTEM= TRUE EQ DATE                   LOCAL TIME= 11 HRS  9 MIN  58.400 SEC
   R  8421.081799          V  6.661165415          C3 -43.31688665          P  128.2455627
*****************************************************************************************************************************
```

Figure 3-7. Sample Output Parameter Report for NOUT=4

to other utilities for processing rather than just printed out. The uses of these forms of output are not discussed in detail in this primer. However, an example is shown in Figure 4-5.

3.1.5 GENERAL FLAGS

Variables in the general flags category include IDEL, IPRNT, and IDYN.

IDEL is a flag that is used in complicated automatic sequences to save core by deleting the propagator and parameter modules. It is not needed for a simple orbit propagation.

IPRNT is a debug printout flag for orbit propagation. It is also unnecessary in a simple orbit propagation.

IDYN is a flag that is used in conjunction with force model changes. If IDYN is not included in the user's input list, GMAS uses the defaulted force model (4-by-4 Earth field, effects of Sun and Moon included). If the user desires to change the force model, he must update the Dynamics File and set IDYN=2 (see Section 3.2 for details).

3.1.6 ORBIT PROPAGATION INPUT EXAMPLES

This section presents primer Problems 2, 3, and 4, which provide examples of various orbit propagation input.

3.1.6.1 Problem 2

PROBLEM: Propagate the orbit of a satellite, given the following:

Epoch:	October 10, 1980; 00:00:00 GMT
Elements:	Keplerian: a = 8525.0 kilometers e = 0.24 i = 68 degrees Ω = 91 degrees ω = 93 degrees T = 248 degrees

Coordinate system of input:	Earth-centered true Earth equator and equinox of date
Propagator:	Time-regularized Cowell (step size = 100 steps per revolution)
Force model:	4-by-4 Earth field, no drag or solar radiation pressure, effects of Moon and Sun included

Stop at first perigee, first apogee, and every 20th perigee and every 20th apogee until October 20, 1980, at 1:30 a.m. At each stop, print out the Cartesian elements (x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$), the position magnitude, the osculating perifocal and apofocal heights, and the period.

USER PROCEDURE: The procedure for solving Problem 2 is as follows:

1. Using Table C-1 of the User's Guide, determine the appropriate value of each variable.

2. For those variables whose default values will not provide the results required for this case, punch out cards with the necessary variables and values, remembering the following:

   a. Each variable name must start after column 1 (the starting point is usually column 3), and the values must not extend beyond column 64.

   b. Each value entered must be followed by a comma.

   c. Real*8 (R*8) type variables (see the Type column in Table C-1 of the User's Guide) must have a decimal point; integer (I*4) type variables have no decimal point.

3. Starting with the complete GMAS deck illustrated in Figure 2-1, replace the cards between the &ORBIN card and the &END card with those punched in step 2. Submit this deck.

AUTOMATIC SEQUENCE: In Figure 2-1, the cards in the deck between the DRIVE1 card and the EOF card constitute the automatic sequence. Generally, the cards before and after the automatic sequence remain the same except for special applications (discussed in subsequent sections). Therefore, the solution to Problem 2 is given (as are the solutions to other problems in this section) as an automatic sequence. The automatic sequence for solving Problem 2 is as follows:

```
col. 1
↓
DRIVE1
PRFCON
ORBINP
 &ORBIN
  IELEM=3,
  ELEM=801010.,0.,
       8525.,.24,68.,
       91.,93.,248.,
  PROPM='TRCOWL',
  H=100.,
  ISTOP=1,998,13,14,13,14,
  STPVAL=801020.,13000.,0.,0.,0.,0.,
  ISKIP=0,0,0,0,19,19,
  IREPT=0,0,0,0,1,1,
  MULTI=0,0,0,0,500,500,
  ITERM=1,
  IPARMS=7,8,9,10,11,12,17,21,22,30,
  NOUT=4,
 &END
ORBIT
EOF
```

The following points should be noted concerning this automatic sequence:

- IELEM=3 because the input elements are Keplerian with true anomaly.

- ICORD is not needed because its default value provides the coordinate system (true Earth equator and equinox of date) required for this case.

- ELEM contains the epoch date and time and the six given Keplerian elements.

- ICENT is not needed because its default value provides the coordinate system center (Earth) required for this case.

- IACORD, SARA, and SDEC are not needed because they pertain to attitude.

- PROPM='TRCOWL' because the time-regularized Cowell propagator is being used.

- IPOPT is not needed because the analytic propagator is not being used.

- H=100 indicates 100 steps per revolution.

- ISTOP, STPVAL, ISKIP, IREPT, and MULTI (the stopping conditions) can be written in the form shown in Table 3-2. Since ITERM=1, use of the values in columns A and B of the table sets the terminal stopping date. Use of the values in column C causes a stop at first perigee. Use of the values in column D causes a stop at first apogee. Use of the values in column E causes a stop at perigee to be skipped 19 times and then accepted once, with the entire process (ISKIP through IREPT) being repeated 500 times. (The MULTI count has been set arbitrarily large to accommodate the propagation.)

- PARMM is not needed because it defaults to 'GPARM'.

- IPARMS=7,8,9,10,11,12,17,21,22,30,. These numbers, taken from Table C-6c of the User's Guide, correspond to the various output information desired.

- NOUT=4 indicates that only the parameters described in the IPARMS array are to be printed at each stop.

- IDEL and IPRNT are not needed.

- IDYN is not needed because its default value provides the force model required for this case.

Table 3-2. Stopping Conditions for Problem 2

VARIABLE	A	B	C	D	E	F
ISTOP	1	998	13	14	13	14
STPVAL	801020.	13000.	0.	0.	0.	0.
ISKIP	0	0	0	0	19	19
IREPT	0	0	0	0	1	1
MULTI	0	0	0	0	500	500

7097/81

OUTPUT: Figure 3-8 shows the printed output resulting from the Problem 2 run.

3.1.6.2 Problem 3: Station Coverage

PROBLEM: Given the following, determine the station coverage of a satellite at the GSFC, Bermuda, Guam, and Hawaii composite antennas for 1 day:

Epoch:	July 29, 1975; 7:09:58.4 GMT
Elements:	Cartesian: x = 6526.439 kilometers y = 0.0 kilometers z = 0.0 kilometers $\dot{x}$ = 0.0 kilometers per second $\dot{y}$ = 3.24286 kilometers per second $\dot{z}$ = 8.026368 kilometers per second
Coordinate system of input:	Earth-centered true Earth equator and equinox of date
Propagator:	Cowell (step size = 100 seconds)
Force model:	4-by-4 Earth field, no drag or solar radiation pressure, effects of Sun and Moon included

USER PROCEDURE: The basic procedure for solving Problem 3 is the same as that specified for Problem 2 (Section 3.1.6.1).

AUTOMATIC SEQUENCE: The automatic sequence for solving Problem 3 is as follows:

```
col. 1
*
DRIVE1
PRFCON
ORBINP
 &ORBIN
  ELEM=750729.,070958.4,
       6526.439,0.,0.,
       0.,3.24286,8.026368,
  H=100.,
  ISTOP=1,41,998,42,998,41,998,42,998,41,998,
        42,998,41,998,42,998,
```

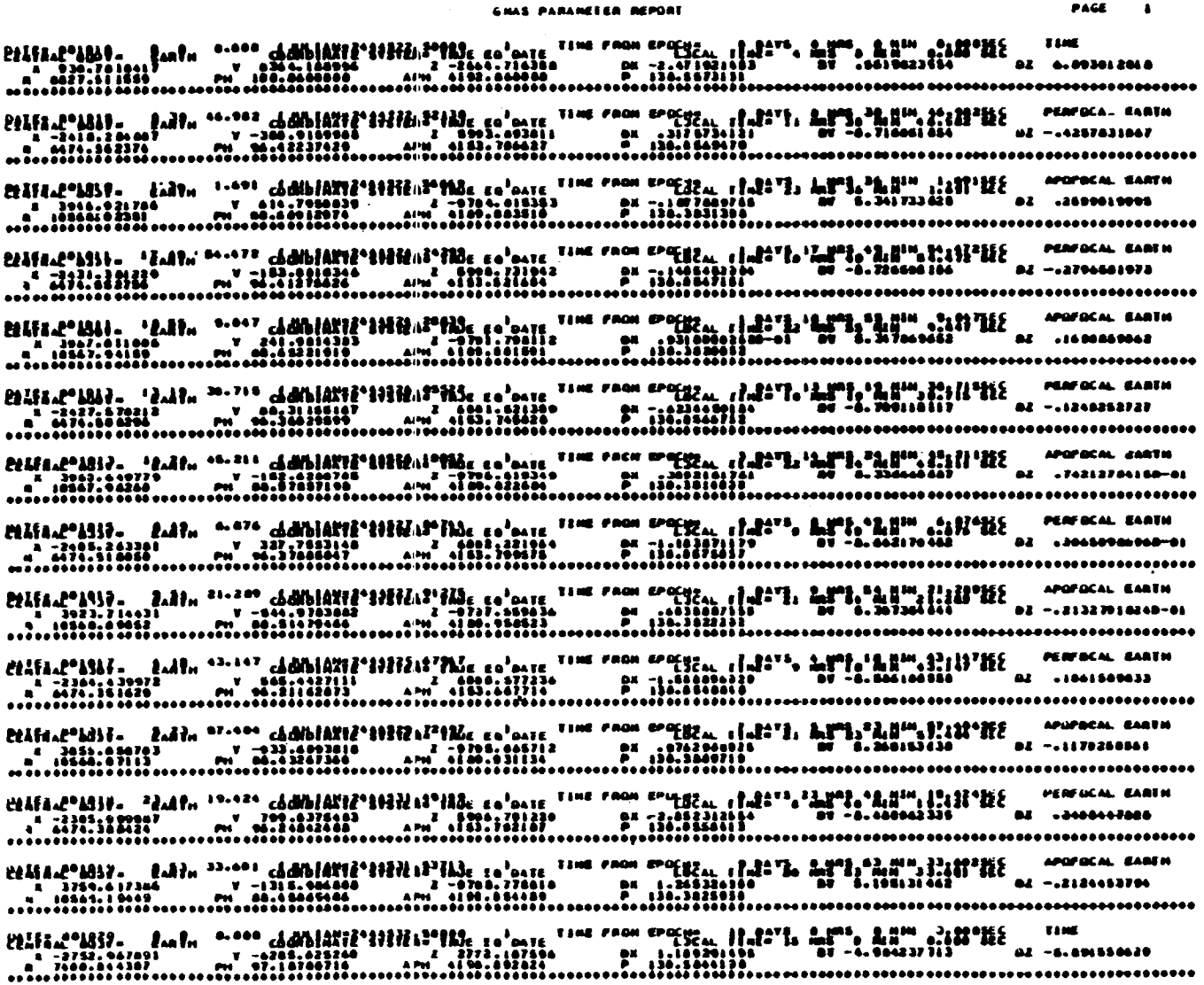


Figure 3-8. Parameter Output Results From Problem 2 Run

```
  STPVAL=86400.,'GSFC/ETC','COMP','GSFC/ETC','COMP',
         'BERMUDA','COMP','BERMUDA','COMP','GUAM',
         'COMP','GUAM','COMP','HAWAII','COMP','HAWAII',
         'COMP',
  IREPT=1,16*50,
  ITERM=1,
 &END
ORBIT
EOF
```

The following points should be noted concerning this automatic sequence:

- IELEM, ICORD, and ICENT are not needed because their default values provide the input elements, coordinate system, and coordinate system center, respectively, required for this case.

- ELEM contains the epoch date and time and six given Cartesian elements.

- PROPM is not needed because its default value provides the propagator (Cowell) required for this case.

- H=100 indicates a 100-second step size.

- The stopping conditions are input as 1 day (86,400 seconds) from epoch; signal acquisitions for the GSFC-composite antenna, Bermuda-composite antenna, Guam-composite antenna, and Hawaii-composite antenna are set; and signal losses for the same station-antenna pairs are set. In addition, the time stop is flagged as terminal, and the other stops are repeated until the time constraint of 1 day is met (i.e., IREPT is set to a value sufficiently large to accommodate at least 1 day of station coverage).

- Each station to be covered requires four positions in the ISTOP array and four corresponding positions in the STPVAL array. The format must be as follows:

```
ISTOP=41,998,42,998,
STPVAL='station name','antenna-name ,
       'station name','antenna-name ,
```

OUTPUT: Figure 3-9 shows the printed output resulting from the Problem 3 run.

3.1.6.3 Problem 4: Shadow Studies

PROBLEM: Given the same orbit propagation information specified in Problem 3, conduct a shadow study for 1 day.

USER PROCEDURE: The basic user procedure for solving this problem is the same as that specified for Problem 2 (Section 3.1.6.1).

AUTOMATIC SEQUENCE: The automatic sequence for solving this problem is as follows:

```
col. 1
↓
DRIVE1
PRFCON
ORBINP
 &ORBIN
  ELEM=750729.,70958.4,
       6526.439,0.,0.,
       0.,3.24286,8.026368,
  H=100.,
  ISTOP=1,50,
  STPVAL=86400.,1.,
  IREPT=1,1000,
  ITERM=1,
 &END
ORBIT
EOF
```

The variables in this automatic sequence are the same as in the Problem 3 automatic sequence except for the stopping conditions. The stopping conditions in this sequence include a terminal stop 1 day from epoch and a stop on all Earth shadow conditions that is repeated 1000 times. (The repeat factor is arbitrarily set to cover the 1-day time span.)

```
                                   GPAS PARAMETER REPORT

STATION: GSFC/ETC ANTENNA: COMP    LAT =  39.00308 DEG  LONG =  283.15567 DEG  ALT =     5.1000 M
STATION: BERMUDA  ANTENNA: COMP    LAT =  32.34926 DEG  LONG =  295.34540 DEG  ALT =   -27.3750 M
STATION: GUAM     ANTENNA: COMP    LAT =  13.31049 DEG  LONG =  144.73782 DEG  ALT =   115.4400 M
STATION: HAWAII   ANTENNA: COMP    LAT =  22.12195 DEG  LONG =  200.33310 DEG  ALT =  1138.1740 M
SIGNAL ACQUIS  GUAM      COMP   DATE 75  7 29  UT   7 58  16.619
SIGNAL LOSS    GUAM      COMP   DATE 75  7 29  UT   8 27  14.442  DURATION  0 D  0 H  28.964 M
SIGNAL ACQUIS  BERMUDA   COMP   DATE 75  7 29  UT   9 23  56.378
SIGNAL ACQUIS  GSFC/ETC  COMP   DATE 75  7 29  UT   9 26  13.338
SIGNAL LOSS    GSFC/ETC  COMP   DATE 75  7 29  UT   9 29  55.639  DURATION  0 D  0 H   3.705 M
SIGNAL LOSS    BERMUDA   COMP   DATE 75  7 29  UT   9 34  33.769  DURATION  0 D  0 H  10.623 M
SIGNAL ACQUIS  BERMUDA   COMP   DATE 75  7 29  UT  14 59  51.700
SIGNAL LOSS    BERMUDA   COMP   DATE 75  7 29  UT  15  9  14.179  DURATION  0 D  0 H   9.375 M
SIGNAL ACQUIS  GSFC/ETC  COMP   DATE 75  7 29  UT  16  9  47.615
SIGNAL ACQUIS  BERMUDA   COMP   DATE 75  7 29  UT  16  9  49.269
SIGNAL LOSS    GSFC/ETC  COMP   DATE 75  7 29  UT  16 10  14.417  DURATION  0 D  0 H   0.447 M
SIGNAL ACQUIS  GSFC/ETC  COMP   DATE 75  7 29  UT  16 16  44.221
SIGNAL LOSS    GSFC/ETC  COMP   DATE 75  7 29  UT  16 24   2.534  DURATION  0 D  0 H   7.305 M
SIGNAL LOSS    BERMUDA   COMP   DATE 75  7 29  UT  16 46  53.093  DURATION  0 D  0 H  37.064 M
SIGNAL ACQUIS  GUAM      COMP   DATE 75  7 29  UT  17 53  49.076
SIGNAL LOSS    GUAM      COMP   DATE 75  7 29  UT  17 56  59.442  DURATION  0 D  0 H   3.173 M
SIGNAL ACQUIS  GSFC/ETC  COMP   DATE 75  7 29  UT  18 19  49.362
SIGNAL ACQUIS  BERMUDA   COMP   DATE 75  7 29  UT  18 21  47.129
SIGNAL LOSS    GSFC/ETC  COMP   DATE 75  7 29  UT  18 46  46.755  DURATION  0 D  0 H  26.957 M
SIGNAL LOSS    BERMUDA   COMP   DATE 75  7 29  UT  18 54  31.896  DURATION  0 D  0 H  32.746 M
SIGNAL ACQUIS  GSFC/ETC  COMP   DATE 75  7 29  UT  20 32  28.793
SIGNAL ACQUIS  BERMUDA   COMP   DATE 75  7 29  UT  20 32  49.480
SIGNAL LOSS    BERMUDA   COMP   DATE 75  7 29  UT  20 39  56.046  DURATION  0 D  0 H   7.113 M
SIGNAL ACQUIS  BERMUDA   COMP   DATE 75  7 29  UT  20 40   3.291
SIGNAL LOSS    GSFC/ETC  COMP   DATE 75  7 29  UT  21  1   5.579  DURATION  0 D  0 H  28.683 M
SIGNAL LOSS    BERMUDA   COMP   DATE 75  7 29  UT  21 14  50.980  DURATION  0 D  0 H  34.755 M
SIGNAL ACQUIS  HAWAII    COMP   DATE 75  7 29  UT  22 42  46.153
SIGNAL LOSS    HAWAII    COMP   DATE 75  7 29  UT  22 53  52.419  DURATION  0 D  0 H  11.104 M
SIGNAL ACQUIS  HAWAII    COMP   DATE 75  7 30  UT   0 51  47.580
SIGNAL LOSS    HAWAII    COMP   DATE 75  7 30  UT   1 31  29.368  DURATION  0 D  0 H  39.690 M
SIGNAL ACQUIS  GUAM      COMP   DATE 75  7 30  UT   3  4   2.411
SIGNAL ACQUIS  HAWAII    COMP   DATE 75  7 30  UT   3 11  38.582
SIGNAL LOSS    GUAM      COMP   DATE 75  7 30  UT   3 36  43.055  DURATION  0 D  0 H  32.677 M
SIGNAL LOSS    HAWAII    COMP   DATE 75  7 30  UT   3 37  32.697  DURATION  0 D  0 H  25.904 M
SIGNAL ACQUIS  GUAM      COMP   DATE 75  7 30  UT   5  9  22.434
SIGNAL LOSS    GUAM      COMP   DATE 75  7 30  UT   5 55  19.863  DURATION  0 D  0 H  45.983 M
```

Figure 3-9. Parameter Output Results From Problem 3 (Station Coverage) Run

OUTPUT: Figure 3-10 shows the printed output resulting from the Problem 4 run.

3.2 FORCE MODEL INPUT

This section discusses the procedures for entering various changes to the GMAS default force model. The GMAS default force model includes a 4-by-4 Earth field and the effects of Moon and Sun. Changes to this force model are accomplished via input cards entered between the PRFCON card and the ORBINP card in the GMAS card deck (see Figure 2-1). Sections 3.2.1 and 3.2.2 specify the methods for entering force model changes for Earth-centered orbits and Moon-centered orbits, respectively. Section 3.2.3 presents primer Problem 5, which provides an example of force model input.

3.2.1 FORCE MODEL CHANGES FOR EARTH-CENTERED ORBITS

Four dynamic forces affect the performance of spacecraft orbiting the Earth:

Dynamic Force	Default
Earth's geopotential effects	4 by 4
Noncentral body perturbations	Moon and Sun
Atmospheric drag	OFF
Solar radiation pressure	OFF

To make changes in the default force model, the user must update the Dynamics File and use the updated file for the propagation. The procedure for this is as follows:

1. Set IDYN=2 under the &ORBIN card. (If this is not done, the default force model will be used.)

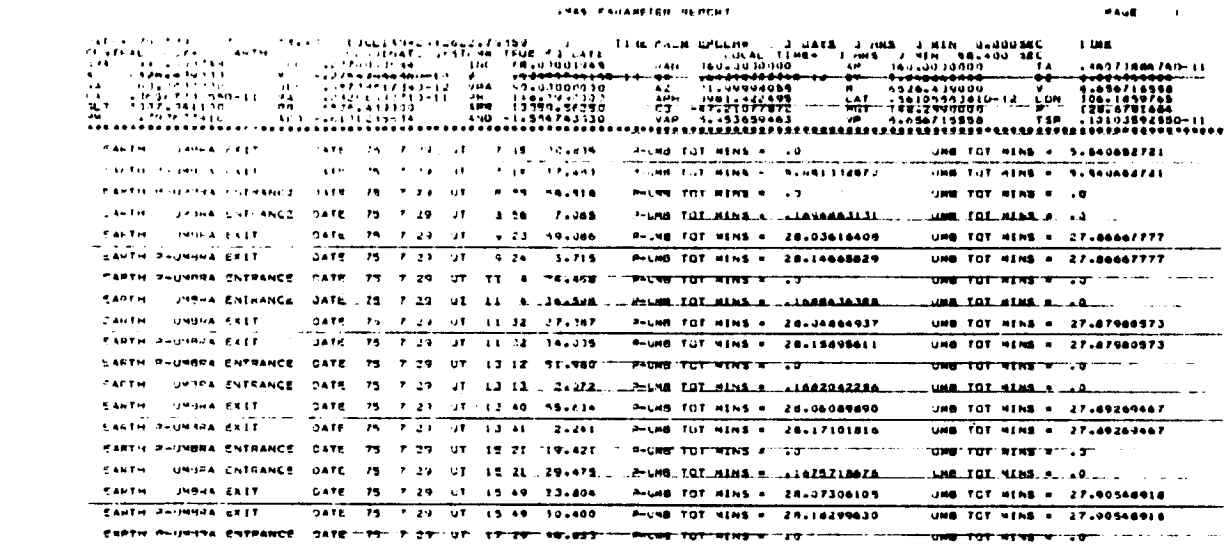


Figure 3-10. Parameter Output Results From Problem 4 (Shadow Studies) Run

2. Include the following cards in the automatic sequence between the PRFCON card and the ORBINP card in the input deck (see Figure 2-1):

```
col. 1
▼
DYNUPD
 &DYNUP
  (variables to be changed)
 &END
 &DYNSEC
  (variables to be changed)
 &END
```

Cards &DYNUP, &END, &DYNSEC, and &END must all be included regardless of whether the user wishes to change variables under them. Variables to be changed are entered in the same fashion as they are under the &ORBIN card.

3. Consult Table 3-3 of this document to determine which variables are to be included. Note the type of each and the card under which each is to be entered (&DYNSEC or &DYNUP).

4. Enter &DYNUP variables under the &DYNUP card and &DYNSEC variables under the &DYNSEC card, remembering that R*8 type variables require decimal points but I*4 type variables do not.

Tables D-2 and D-3 in Appendix D of the User's Guide provide descriptions and default values for all variables under &DYNUP and &DYNSEC, respectively. Most of these variables are fixed constants (e.g., π, GM) or are not usually changed (e.g., harmonics coefficients). Table 3-3 of this primer presents the most frequently used dynamics variables from Tables D-2 and D-3 of the User's Guide. The variables are presented in this table according to the four categories specified at the beginning of this section.

Table 3-3. Frequently Used Dynamics Variables

VARIABLE	DIMENSION	TYPE	NAMELIST	DESCRIPTION	DEFAULT
			EARTH'S GEOPOTENTIAL EFFECTS		
MAXOR1	1	I*4	DYNSEC	MAXIMUM ORDER OF POTENTIAL FIELD TO BE USED (FIELD IS MAXDE1 x MAXOR1)	4
MAXDE1	1	I*4	DYNSEC	MAXIMUM DEGREE OF POTENTIAL FIELD TO BE USED	4
IRESN1	1	I*4	DYNSEC	RESONANCE POTENTIAL SWITCH: = 1, INCLUDE = 2, DO NOT INCLUDE	2
IPM1	1	I*4	DYNSEC	POLAR MOTION SWITCH: = 1, INCLUDE = 2, DO NOT INCLUDE	2
			NONCENTRAL BODY PERTURBATIONS		
INC1	7	I*4	DYNSEC	NONCENTRAL BODY INDICATOR = 1, EARTH = 2, MOON = 3, SUN = 4, MARS = 5, JUPITER = 6, SATURN = 7, URANUS = 8, NEPTUNE = 9, PLUTO =10, MERCURY =11, VENUS	2, 3, 0, 0, 0, 0, 0
			ATMOSPHERIC DRAG		
IDRAG1	1	I*4	DYNSEC	ATMOSPHERIC DRAG SWITCH: = 1, HARRIS-PRIESTER DRAG = 2, NO DRAG = 3, JACCHIA-ROBERTS DRAG MODEL	2
AREA	1	R*8	DYNUP	AREA OF SPACECRAFT (SQUARE KILOMETERS)	$.22 \times 10^{-5}$
SCMASS	1	R*8	DYNUP	MASS OF SPACECRAFT (KILOGRAMS)	200.
CSUBDZ	1	R*8	DYNUP	DRAG COEFFICIENT	2.0
RHO1	1	R*8	DYNUP	VARIATION IN DRAG COEFFICIENT	0.0
			SOLAR RADIATION PRESSURE		
ISP1	1	I*4	DYNSEC	SOLAR RADIATION PRESSURE SWITCH. = 1, INCLUDE = 2, DO NOT INCLUDE	2
AREA	1	R*8	DYNUP	DESCRIBED ABOVE	$.22 \times 10^{-5}$
SCMASS	1	R*8	DYNUP	DESCRIBED ABOVE	200.
CSUBR	1	R*8	DYNUP	REFLECTIVITY CONSTANT	1.2

7907/81

3.2.2 FORCE MODEL CHANGES FOR MOON-CENTERED ORBITS

Although GMAS is primarily used for the propagation of Earth-centered orbits, it does have the capability to propagate non-Earth-centered orbits. To propagate a Moon-centered orbit, the user follows the normal procedure for entering force model changes, with the following additions:

1. Include IBODY1=2 under the &DYNSEC card. (The default value for IBODY1 is 1 (Earth).)
2. Include INC1=1,3 under the &DYNSEC card. (This is optional, but the effects of Earth and Sun should be included as noncentral body perturbations, and the effects of the Moon must be removed.)
3. Include ICENT=2 under the &ORBIN card if the input elements are Moon centered.

The following points should be noted concerning force model changes for Moon-centered orbits:

- The same variables used to make changes to the Earth potential field (MAXDE1, MAXOR1) are used to make changes to the lunar potential field (whose default size is 4 by 4).
- Atmospheric drag is not a dynamic force with respect to the Moon.
- The default output is with respect to the integrator central body--in this case, the Moon. If a change is desired, variable NOUT(3) must be changed under the &ORBIN card.

3.2.3 FORCE MODEL INPUT EXAMPLE

This section presents primer Problem 5, which provides an example of satellite orbit propagation requiring default force model changes.

PROBLEM 5: Given the following, propagate a satellite orbit for 5 days, printing level 1 output every 12 hours:

Epoch:	October 10, 1980, 00:00:00 GMT
Elements:	Keplerian: a = 8525.0 kilometers e = 0.14 i = 68 degrees Ω = 91 degrees ω = 93 degrees T = 248 degrees
Coordinate system of input:	Earth-centered true Earth equator and equinox of date
Propagator:	Time-regularized Cowell (step size = 100 steps per revolution)
Force model:	8-by-8 Earth field, effects of Sun and Moon included, resonance effects included, no solar radiation, Harris-Preister drag model with 2.2 drag coefficient and 0.02 drag coefficient variation; spacecraft mass of 150 kilograms, spacecraft area of 0.5 x 10^5 square kilometers

AUTOMATIC SEQUENCE: The automatic sequence for solving Problem 5 is as follows:

```
col. 1
↓
DRIVE1
PRFCON
DYNUPD
 &DYNUP
  AREA=.5D-5,SCMASS=150.,
  CSUBDZ=2.2,RHO1=.02,
 &END
 &DYNSEC
  MAXOR1=8,MAXDE1=8,IRESN1=1,
  IDRAG1=1,
 &END
ORBINP
 &ORBIN
  IELEM=3
  ELEM=801010.0,0.,
       8525.,.14,68.,
       91.,93.,248.,
  PROPM='TRCOWL',H=100.,
  ISTOP=1,STPVAL=43200.,
```

```
  IREPT=10,
  NOUT=1,
  IDYN=2,
 &END
ORBIT
EOF
```

The following points should be noted concerning this automatic sequence:

- Because force model changes (Dynamics File updates) are required, the DYNUPD card must be included between the PRFCON card and the ORBINP card.

- The DYNUPD card must be followed by the &DYNUP card. The &DYNUP card must be followed by the &DYNUP variables to be changed (see Table 3-3), which in this case include spacecraft area (AREA), spacecraft mass (SCMASS), drag coefficient (CSUBDZ), and drag coefficient variation (RHO1). The &DYNUP variables must be followed by an &END card.

- The first &END card must be followed by the &DYNSEC card. The &DYNSEC card must be followed by the &DYNSEC variables to be changed (see Table 3-3), which in this case include order and degree of potential field (MAXOR1 and MAXDE1) and switches for resonance potential (IRESN1) and atmospheric drag (IDRAG1). Solar radiation pressure (ISP1) is not included because its default value (2) provides the required result. The noncentral body indicator (INC1) is not included because its default value (2,3) provides the required result (Moon and Sun). The polar motion switch (IPM1) is not included since polar motion is not considered in this case. The &DYNSEC variables must be followed by an &END card.

- Variables are entered under the ORBINP card as in Problem 2 (Section 3.1.6.1).

- The stopping logic is as follows: propagation stops on time (ISTOP=1) every 12 hours (STPVAL=43200.), and this is repeated 20 times (IREPT=20) for a total of 10 days.

- NOUT=1 because level 1 output is desired.

- IDYN=2 because the Dynamics File is being updated.

The following additional points should be noted:

- Both &DYNUP and &DYNSEC must be included even if no variable changes are required for one of them.

- The following order of cards must be adhered to in an orbit propagation with force model changes:

```
col. 1
*
DYNUPD
 &DYNUP
  (variables)
 &END
 &DYNSEC
  (variables)
 &END
ORBINP
 &ORBIN
  (variables)
 &END
ORBIT
```

- The order of the variables under various & cards (NAMELIST cards) is flexible. The variables are usually listed in the same order as they are found in tables.

- It is not necessary to use one card for each variable as has been done in previous examples. It is acceptable to include several variables and their values (separated by commas) on the same line (e.g., MAXOR1=8,MAXDE1=8,IRESN1=1,). However, the last variable value on each line must be followed by a comma, and no GMAS input can extend beyond column 64.

- Variable values in scientific notation such as 0.5×10^{-5} must be expressed in double-precision exponential form (.5D-5).

- When including changes to the force model, the user must set IDYN=2 under the &ORBIN card.

OUTPUT: Figure 3-11 shows the printed output resulting from the Problem 5 run.

3.3 STEPS IN PROPAGATING AN ORBIT WITH GMAS: A SUMMARY

As a brief summary of the material covered in Sections 3.1 and 3.2, this section specifies the step-by-step procedure for propagating a satellite orbit using GMAS. The procedure is as follows:

1. Develop an automatic sequence by following the steps specified below. The column number specified in parentheses indicates the column in which the card's first letter must be punched.

   a. Start with DRIVE1 (column 1) followed by PRFCON (column 1).

   b. If the force model is not to be changed, go to step h.

   c. Determine the appropriate values of each variable in Table 3-3. Note whether the variables are under &DYNUP or &DYNSEC.

   d. Include DYNUPD (column 1) followed by &DYNUP (column 2).

   e. Include the appropriate dynamics variables (if any) between columns 2 and 64.

   f. Include &END (column 2) followed by &DYNSEC (column 2).

DATE= 80/10/10 0 EARTH 0.000 (JULIAN=2444522.500000) TIME FROM EPOCH= 0 DAYS 0 HRS 0 MIN 0.000SEC TIME
CENTRAL BODY= EARTH COORDINATE SYSTEM= TRUE EQ DATE LOCAL TIME= 4 HRS 0 MIN 0.000 SEC
SMA 8525.000000 ECC .1400000000 INC 68.00000000 RAN 91.00000000 AP 93.00000000 TA 240.0000000

DATE= 80/10/10 0 EARTH 0.000 (JULIAN=2444522.500000) TIME FROM EPOCH= 0 DAYS 12 HRS 0 MIN 0.000SEC TIME
CENTRAL BODY= EARTH COORDINATE SYSTEM= TRUE EQ DATE LOCAL TIME= 17 HRS 0 MIN 0.000 SEC
SMA 8525.200270 ECC .1400183170 INC 68.00054469 RAN 90.28968316 AP 92.88059835 TA 105.6610794

DATE= 80/10/11 0 EARTH 0.000 (JULIAN=2444523.500000) TIME FROM EPOCH= 1 DAYS 0 HRS 0 MIN 0.000SEC TIME
CENTRAL BODY= EARTH COORDINATE SYSTEM= TRUE EQ DATE LOCAL TIME= 4 HRS 0 MIN 0.000 SEC
SMA 8526.477560 ECC .1401194248 INC 68.00208962 RAN 89.59624857 AP 92.37091063 TA 260.4776119

DATE= 80/10/11 12 EARTH 0.000 (JULIAN=2444524.000000) TIME FROM EPOCH= 1 DAYS 12 HRS 0 MIN 0.000SEC TIME
CENTRAL BODY= EARTH COORDINATE SYSTEM= TRUE EQ DATE LOCAL TIME= 17 HRS 0 MIN 0.000 SEC
SMA 8523.197706 ECC .1399960198 INC 67.99843164 RAN 88.88514858 AP 92.22365622 TA 117.8171673

DATE= 80/10/12 0 EARTH 0.000 (JULIAN=2444524.500000) TIME FROM EPOCH= 2 DAYS 0 HRS 0 MIN 0.000SEC TIME
CENTRAL BODY= EARTH COORDINATE SYSTEM= TRUE EQ DATE LOCAL TIME= 5 HRS 0 MIN 0.000 SEC
SMA 8527.039620 ECC .1403175797 INC 68.00274506 RAN 88.19303638 AP 91.80436511 TA 273.7008488

DATE= 80/10/12 12 EARTH 0.000 (JULIAN=2444525.000000) TIME FROM EPOCH= 2 DAYS 12 HRS 0 MIN 0.000SEC TIME
CENTRAL BODY= EARTH COORDINATE SYSTEM= TRUE EQ DATE LOCAL TIME= 18 HRS 0 MIN 0.000 SEC
SMA 8521.187936 ECC .1401025906 INC 67.99592234 RAN 87.47969079 AP 91.57414460 TA 129.3337867

DATE= 80/10/13 0 EARTH 0.000 (JULIAN=2444525.500000) TIME FROM EPOCH= 3 DAYS 0 HRS 0 MIN 0.000SEC TIME
CENTRAL BODY= EARTH COORDINATE SYSTEM= TRUE EQ DATE LOCAL TIME= 5 HRS 0 MIN 0.000 SEC
SMA 8526.039527 ECC .1404706464 INC 68.00143133 RAN 86.75951155 AP 91.32592565 TA 287.7361074

DATE= 80/10/13 12 EARTH 0.000 (JULIAN=2444526.000000) TIME FROM EPOCH= 3 DAYS 12 HRS 0 MIN 0.000SEC TIME
CENTRAL BODY= EARTH COORDINATE SYSTEM= TRUE EQ DATE LOCAL TIME= 18 HRS 0 MIN 0.000 SEC
SMA 8519.415289 ECC .1402951286 INC 67.99341556 RAN 86.07332361 AP 90.95783345 TA 140.2951676

DATE= 80/10/14 0 EARTH 0.000 (JULIAN=2444526.500000) TIME FROM EPOCH= 4 DAYS 0 HRS 0 MIN 0.000SEC TIME
CENTRAL BODY= EARTH COORDINATE SYSTEM= TRUE EQ DATE LOCAL TIME= 5 HRS 0 MIN 0.000 SEC
SMA 8527.965986 ECC .1404090885 INC 67.99797829 RAN 85.38441373 AP 90.91753941 TA 302.0477356

DATE= 80/10/14 12 EARTH 0.000 (JULIAN=2444527.000000) TIME FROM EPOCH= 4 DAYS 12 HRS 0 MIN 0.000SEC TIME
CENTRAL BODY= EARTH COORDINATE SYSTEM= TRUE EQ DATE LOCAL TIME= 18 HRS 0 MIN 0.000 SEC
SMA 8517.993844 ECC .1405175152 INC 67.99118727 RAN 84.66615191 AP 90.38496969 TA 150.8150489

DATE= 80/10/15 0 EARTH 0.000 (JULIAN=2444527.500000) TIME FROM EPOCH= 5 DAYS 0 HRS 0 MIN 0.000SEC TIME
CENTRAL BODY= EARTH COORDINATE SYSTEM= TRUE EQ DATE LOCAL TIME= 6 HRS 0 MIN 0.000 SEC
SMA 8518.080564 ECC .1403276170 INC 67.99294674 RAN 83.97634517 AP 90.49797150 TA 318.4859524

Figure 3-11. Parameter Output Results From Problem 5 Run

g. Include the appropriate dynamics variables (if any) between columns 2 and 64, followed by &END (column 2).

h. Include ORBINP (column 1) followed by &ORBIN (column 2).

i. Using Table C-1 in the User's Guide, determine the appropriate value of each variable. Enter these values starting in column 2.

j. If force model changes were included in steps c through g, include variable IDYN=2.

k. Follow the variables with &END (column 2), ORBIT (column 1), and EOF (column 1).

2. Starting with the card deck shown in Figure 2-1, replace the automatic sequence part of the deck with the new automatic sequence created in step 1. Submit this deck.

SECTION 4 - AUTOMATIC SEQUENCES

The reader, having studied Sections 1, 2, and 3 of this primer, is now able to propagate a satellite orbit using GMAS. He may, however, be questioning at this point whether the complex procedure used to enter information into GMAS is really necessary. Other software packages seem much more straightforward. Why not simply read data into GMAS in the form of a card stating the orbit propagation objective, followed by a number of cards specifying all the necessary parameter information?

The answer to this is that GMAS input data is not simply data. It is, rather, a high-level programming language. This language, called the automatic sequence, is a step-by-step procedure for performing various tasks in a logical way, allowing the passing of information from one task to another.

This section discusses the GMAS input language. Section 4.1 describes the various types of cards used in GMAS automatic sequences. Section 4.2 provides examples of the use of these cards.

4.1 AUTOMATIC SEQUENCE CARDS

Automatic sequences begin with the DRIVE1 and PRFCON cards and end with an EOF card, except in some special cases (see Section 5). The GMAS automatic sequence is conceptually much easier if these three cards are assumed to be permanent fixtures in it. Therefore, all automatic sequences described in this section are assumed to begin with the DRIVE1 and PRFCON cards and end with an EOF card. The cards situated between the PRFCON card and the EOF card are called

GMAS automatic sequence cards. GMAS automatic sequence cards are of five basic types:

Card Type	Start Column
Utility	1
NAMELIST input	2
Dynamic array	1
Logical directive	1
Comment	1

Utility cards represent tasks that are to be accomplished (e.g., propagating an orbit, creating a graph).

NAMELIST input cards are easily recognized. The first NAMELIST card in a deck contains an ampersand (&) in column 2 followed by a NAMELIST name (e.g., &DYNUP, &DYNSEC, &ORBIN). This first card is followed by cards containing a collection of variables and their values separated by commas, beginning in column 2 and extending no farther than column 64. These cards are followed by an &END card, which begins in column 2. These NAMELIST input cards are for user input to the utilities.

Dynamic array cards are used to create or delete information storage areas called dynamic arrays. These arrays are used to pass information between utilities.

Logical directive cards direct the flow of the automatic sequence. The reader familiar with any computer programming languages will recognize the logical directives as the branch (GO TO) statements and IF tests used in a program.

Comment cards have no effect on the operation of the automatic sequence, but they are often useful for description and clarification.

The following subsections describe these five basic types of automatic sequence cards.

4.1.1 UTILITY CARDS

Beginning with the simple automatic sequence described in the solution to Problem 5 (Section 3.2.3), but stripping away the NAMELIST input cards, the following cards remain:

```
DRIVE1
PRFCON
DYNUPD
ORBINP
ORBIT
EOF
```

The three cards between PRFCON and EOF are all utilities in this case. DYNUPD is a utility that updates the force model (Dynamics File) to be used by the propagator. ORBINP, the orbit input processor utility, sets up all the propagation information necessary to propagate the orbit. ORBIT actually propagates the orbit and prints out the results.

GMAS has a number of available utilities in addition to the three specified above. Table 4-1 lists some of the currently available GMAS utilities. It should be noted that the GMAS user is not restricted to GMAS utilities; he can create his own utilities for his own mission-specific problems. The number of available utilities in this case is virtually unlimited.

4.1.1.1 Example of Utility Card Use

As a simple example of utility card use, this section presents the following case, in which the objective is to perform the following tasks in the order specified:

1. Update the force model (Dynamics File).
2. Process input for the orbit propagator.
3. Propagate an orbit.
4. Do some arithmetic on some existing dynamic arrays.
5. Print out dynamic array values.

Table 4-1. GMAS Utilities

UTILITY	NAMELIST	DESCRIPTION	DETAILED DESCRIPTION SOURCE
ARITH	ARITH	DYNAMIC ARRAY ARITHMETIC UTILITY	SOFTWARE RESOURCES, SECTION 2.1
AVECON	AVECO	CONVERTS OSCULATING ELEMENTS TO AVERAGED ELEMENTS FOR PROPAGATION BY AVERAGED ORBIT PROPAGATOR (AVGVOP)	USER'S GUIDE, SECTION 5
DELTAV	DELTA	ESTIMATES EFFECTS ON AN ORBIT DUE TO AN IMPULSIVE BURN MANEUVER	SOFTWARE RESOURCES, SECTION 2.2
DYNUPD	DYNUP DYNSEC	UPDATES DYNAMICS USED IN PROPAGATION OF AN ORBIT	USER'S GUIDE, SECTION 5
ORBINP	ORBIN	PROCESSES INPUT INFORMATION TO BE USED FOR PROPAGATION OF AN ORBIT	USER'S GUIDE, SECTION 5
ORBIT	–	PROPAGATES AN ORBIT	USER'S GUIDE, SECTION 5
PLTDYN	PLTDY	GENERATES PLOTS OF DYNAMIC ARRAYS ON A CARTESIAN GRID	USER'S GUIDE, SECTION 5
PREPLT	PREPL	PREPARES DYNAMIC ARRAYS PRODUCED BY AN ORBIT PROPAGATION FOR PLOTTING BY PLTDYN; USED EXTENSIVELY IN ORBIT COMPARISON GRAPHS	SOFTWARE RESOURCES, SECTION 2.4
PRTDYN	PRTDY	PRINTS VALUES OF DYNAMIC ARRAYS	USER'S GUIDE, SECTION 5

7907/81

In this case, the required utility cards are placed in the automatic sequence (between PRFCON and EOF) as follows (see Table 4-1):

```
DRIVE1
PRFCON
DYNUPD
ORBINP
ORBIT
ARITH
PRTDYN
EOF
```

4.1.1.2 Core Requirements

When the computer starts running GMAS, it reserves a certain memory storage area, called a region, for the program. The size of this region, or core, is specified by the user on the // EXEC GMAS,REGION.GO=376K card (see Figure 2-1).

As GMAS proceeds through the automatic sequence, it must bring utilities into its region in order to execute them (see Figure 4-1). If no more room is left in its region for a utility that is needed, GMAS terminates. Therefore, it is important to keep the region "clean" (i.e., to delete from the region utilities that are no longer being used).

Table 3-3 of the User's Guide specifies the format of the utility card. The user can include different combinations of the letters L, X, D, and K after the utility name to instruct GMAS on the handling of the utility. L tells GMAS to load the utility in the GMAS region from the disk storage space. X tells it to execute the utility. D tells it to delete the utility from the region after execution. K tells it to keep the utility in the region after execution. For example, AVECON,LXK tells GMAS to load, execute, and keep the AVECON utility. Status flags (L, X, D, and K) need not always be included after a utility name, since the absence of status flags after a name causes the computer to load, execute, and delete the utility after execution (i.e. default = LXD).

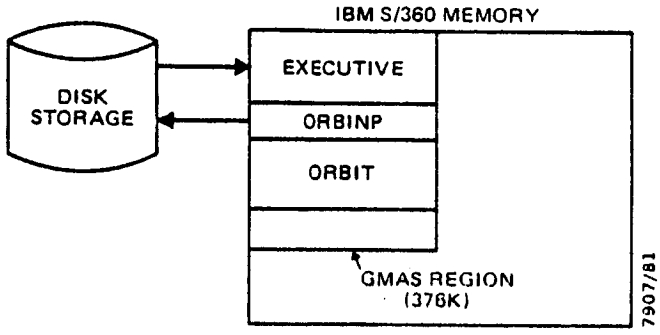


Figure 4-1. Accessing Utilities

4.1.2 NAMELIST INPUT CARDS

NAMELIST input is the primary means of communication between the user and a utility. Most utilities require NAMELIST input (see Table 4-1). The rules for NAMELIST input are as follows:

- NAMELIST input follows the utility card that is using it.
- On the first card, the symbol & must be in column 2, followed by the NAMELIST name.
- The next card must contain, starting in column 2 and extending no farther than column 64, a list of variables and their values. Each value must be followed immediately by a comma. It is to be noted that blanks between the value and the comma are treated as zeros (i.e., IELEM=1 , is interpreted as IELEM=10,).
- Values of variables that are designated as R*8 variables must contain decimal points. I*4 variables cannot have decimal points.
- The NAMELIST must terminate with &END, either on a separate card or following the comma that completes the last data item.

The following is an example of NAMELIST input:

```
col. 2
↓
&INPUT
 ISWTCH=1,IFLAG=3,ITERO=1,17*0,
 STATE=200.,5000.,2.D4,2.D-5,1.D-4,2.D-4,
&END
```

The following points should be noted regarding the example:

- ITERO is an 18-element array. Instead of entering 17 zeros on the card, it is acceptable to write 17*0.

- Every line of NAMELIST variable input must end with a comma.
- Scientific notation must be replaced by double-precision exponential notation (e.g., 3×10^4 = 3.D4, 2×10^{-5} = 2.D-5).

The automatic sequence example presented in Section 4.1.1.1 is shown below with the corresponding NAMELIST input (see Table 4-1):

```
col. 1
*
DRIVE1
PRFCON
DYNUPD
 &DYNUP
  (variables)
 &END
 &DYNSEC
  (variables)
 &END
ORBINP
 &ORBIN
  (variables)
 &END
ORBIT
ARITH
 &ARITH
  (variables)
 &END
PRTDYN
 &PRTDY
  (variables)
 &END
EOF
```

NAMELIST variables are a feature of the IBM version of FORTRAN. Reference 6 provides a complete description of this feature.

4.1.3 DYNAMIC ARRAY CARDS

Dynamic arrays can be thought of as variable arrays in the automatic sequence. They serve primarily as a means of communication between utilities. To create a dynamic array, the user must specify the array name, the type of variable,

and the dimensions. Table 3-4 of the User's Guide specifies the general format of the dynamic array allocation card. NAMELIST VALUE can be used to input values into a dynamic array. Table 3-5 of the User's Guide gives the rules for the use of NAMELIST VALUE.

Dynamic arrays use space in the GMAS region. Therefore, just as with utilities, it is preferable to keep the region free of dynamic arrays that are not being used (see Section 4.1.1.2). To delete a dynamic array from the GMAS region, the user employs the deallocation card. Table 3-6 of the User's Guide specifies the format of the deallocation card.

4.1.3.1 Examples of Dynamic Array Allocation

Tables 3-4 through 3-6 of the User's Guide specify the dynamic array card formats. Two examples of dynamic array allocation are provided below.

EXAMPLE 1: The objective is to allocate a one-dimensional, five-element, R*8 array named ALPHA and to fill the array with the numbers .1, .2, .3, .4, .5. The dynamic array cards required to accomplish this are as follows:

```
col. 1
'
*ALPHA,A,R8,1,5
 &VALUE
  D=.1,.2,.3,.4,.5,
 &END
```

The following points should be noted regarding these cards:

- On the first card, * denotes that a dynamic array is to be allocated; ALPHA is the name of the dynamic array (maximum length = six characters); A denotes allocation; 1 denotes a one-dimensional array; and 5 denotes five-element length.

- To assign values to dynamic array ALPHA, the allocation (*) card must be followed directly with a NAMELIST VALUE card, which in turn must be followed by a card on which D is set equal to the values in the array. Array ALPHA is therefore stored as follows:

```
ALPHA(1)=.1
ALPHA(2)=.2
ALPHA(3)=.3
ALPHA(4)=.4
ALPHA(5)=.5
```

EXAMPLE 2: The objective is to allocate a two-dimensional I*4 array named BETA, where BETA is the 2-by-3 matrix

$$\text{BETA} = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$$

The dynamic array cards necessary to accomplish this are as follows:

```
col. 1
*BETA,A,I4,2,2,3
 &VALUE
  D=1,2,3,4,5,6,
 &END
```

It should be noted that although BETA is a two-dimensional array, variable D cannot be multidimensional. To initialize multidimensional dynamic arrays, the user must index variable D to refer to the corresponding position in a one-dimensional array. Multidimensional arrays are stored with the first index increasing most rapidly, followed by the second, and so forth. A general formula to relate this is as follows: Given an M-by-N-by-P array, A, element A(x,y,z) may be referenced by the single subscript i, where

$$i = (z-1)MN + (y-1)M + x$$

4.1.3.2 Rules for Positioning Dynamic Arrays in Automatic Sequences

The following are rules for positioning dynamic arrays in automatic sequences:

- Dynamic array names must be less than or equal to six characters in length. VARIABLE is an illegal name; VARIAB is acceptable.
- Dynamic arrays allocated in the beginning of the automatic sequence must be placed directly after the DRIVE1 card or the PRFCON card. NAMELIST VALUE must directly follow the allocation card of the dynamic array that it initializes.
- Cards relating to a specific utility should be placed directly after the utility card and should be in the following order:
  - Dynamic array allocation statements and their initialization NAMELISTs--These dynamic arrays will be allocated and initialized before the execution of the utility.
  - Utility input data (through NAMELIST)--These data will be read in during the execution of the utility.
  - Dynamic array deallocation statements--These dynamic arrays will be deallocated after the execution of the utility.

These rules are illustrated with the following hypothetical example:

```
DRIVE1
PRFCON
*INDEX,A,I4,1,1
UTIL1
UTIL2
*ARR1,A,R8,1,3
 &VALUE
```

```
  D=.2,.3D5,.4,
 &END
*ARR2,A,R8,1,3
 &UTIL
  X=1.3,I=2,
 &END
**ARR2,INDEX
UTIL3
   :
```

In the automatic sequence presented above, actions occur in the following order:

1. The dynamic variable INDEX is allocated.
2. UTIL1, a utility with no user input, is executed.
3. ARR1 is allocated and initialized.
4. ARR2 is allocated.
5. UTIL2 is executed. During its execution, the NAMELIST UTIL variables are read in.
6. After UTIL2 execution, dynamic arrays ARR2 and INDEX are deleted. ARR1 remains allocated.

4.1.4 LOGICAL DIRECTIVE CARDS

Logical directive cards are of four types: LABEL, LOOP TO, GO TO, and IF ... GO TO. These cards must immediately precede a utility card or the EOF card. The following sections describe the functions of the logical directive cards and provide examples of their use.

4.1.4.1 LABEL XXXXXX

The LABEL card is used in conjunction with the other three logical directive cards. It provides a reference in the automatic sequence to which the logical directive cards are directed. Table 3-9 of the User's Guide specifies the format of the LABEL card.

4.1.4.2 LOOP TO XXXXXX,I

Segments of an automatic sequence can be repeated (I) times through use of the LOOP TO and LABEL cards. Table 3-8 of the User's Guide describes the general format of the LOOP TO card.

The following portion of an automatic sequence illustrates the use of the LOOP TO card:

```
  :
LOOP TO BOTTOM,3
UTIL1
ORBINP
 &ORBIN
  (variables)
 &END
ORBIT
LABEL BOTTOM
UTIL2
  :
```

Use of the preceding cards in an automatic sequence causes the portion of the sequence from UTIL1 through ORBIT to be repeated three times. Execution continues with UTIL2.

4.1.4.3 GO TO XXXXXX

The GO TO statement is used for unconditional transfer of program operation to another point in an automatic sequence. In the statement GO TO XXXXXX, XXXXXX represents the name given on the LABEL statement, which specifies the point at which execution continues. Table 3-10 of the User's Guide describes the general format of the GO TO card.

The following portion of an automatic sequence illustrates the use of the GO TO card:

```
  :
GO TO BOTTOM
UTIL1
ORBINP
```

```
 &ORBIN
  (variables)
 &END
ORBIT
LABEL BOTTOM
UTIL2
   :
   :
```

Use of the preceding cards in an automatic sequence results in the bypassing of utilities UTIL1, ORBINP, and ORBIT. Sequence execution jumps from the GO TO BOTTOM card to the LABEL BOTTOM card, and execution continues with UTIL2.

4.1.4.4 IF (X.op.Y) GO TO XXXXXX

The IF statement allows an element of a dynamic array (X) or a dynamic variable[1] (X) to be logically tested against another element of a dynamic array (Y), a dynamic variable (Y), or a constant (Y). Dynamic arrays used in testing must be one dimensional. If test conditions are satisfied, control is transferred to a LABEL statement containing the name XXXXXX. Table 3-11 of the User's Guide describes the general format of the IF card.

In the following example of IF card use, ICOUNT, ITOTAL, and VEL are dynamic arrays that are allocated as follows:

```
*ICOUNT,A,I4,1,1
*ITOTAL,A,I4,1,1
*VEL,A,R8,1,3
```

Therefore, ICOUNT and ITOTAL are one-element I*4 dynamic arrays, and VEL is a three-element R*8 array. These arrays are assumed to contain the following values: ICOUNT=1, ITOTAL=4, and VEL=5.,4.,3..

[1] A dynamic variable is a one-element dynamic array.

The following three IF GO TO tests are applied:

```
      :
      :
IF (ICOUNT.GT.0) GO TO END
UTIL1
UTIL2
      :
      :
LABEL END
      :
      :
```

Since ICOUNT=1 is greater than 0, automatic sequence execution skips to LABEL END.

```
      :
      :
IF (ICOUNT.GE.ITOTAL) GO TO 50
UTIL1
UTIL2
      :
      :
LABEL 50
      :
      :
```

Since ICOUNT=1 and TOTAL=5, the test is not satisfied; therefore, the automatic sequence continues executing UTIL1, UTIL2, and so forth.

```
      :
      :
IF (VEL(3).LE.3.) GO TO TOWN
      :
      :
LABEL TOWN
      :
      :
```

Since the third element of the VEL array is 3., the test is satisfied, and automatic sequence execution skips to LABEL TOWN.

4.1.5 COMMENT CARDS

Each comment card begins with the symbol + in its first column. Table 3-7 of the User's Guide describes the comment card. Comments can appear anywhere in an automatic sequence except within the bounds of a NAMELIST (i.e., between &NAME and &END). They have no effect on the execution of the automatic sequence. Comments (and all GMAS input) are truncated beyond card column 64.

The following portion of an automatic sequence illustrates the use of the comment card:

```
   :
   :
+PROCESS ORBIT INPUT INFORMATION
ORBINP
 &ORBIN
  (variables)
 &END
+PROPAGATE THE ORBIT
ORBIT
+THIS IS THE END
EOF
```

4.2 EXAMPLES OF AUTOMATIC SEQUENCE CARD USE

This section provides two examples of automatic sequences using the various types of cards described in Section 4.1. Although the first example is not a typical satellite mission problem, it illustrates the use of logical directives in an automatic sequence. It also illustrates the power of the automatic sequence and the fact that the sequence is not just "data." The second problem demonstrates how looping and testing procedures can be incorporated into an automatic sequence in the solution of a satellite mission problem.

4.2.1 PROBLEM 6

PROBLEM: Given 10 numbers in a dynamic array (NUMBER), determine the smallest number and print it out along with the numbers.

AUTOMATIC SEQUENCE: Although a simple FORTRAN program could be used to solve this problem, the GMAS automatic sequence language is used here for solution. The automatic sequence presented below solves Problem 6. Explanations are included to clarify the actions implied by the automatic sequence.

Automatic Sequence	Explanation
DRIVE1	
PRFCON	
+	Comments
+INPUT 10 NUMBERS IN NUMBER DYNAMIC	
+ARRAY	
+	
*NUMBER,A,R8,1,10	Allocate necessary
&VALUE	dynamic arrays
D=5.,7.,11.,8.,10.,25.,.7,.9,3.,58.,	
&END	
*WORK,A,R8,1,10	
*SMALL,A,R8,1,1	
+	Comments
+DETERMINE THE SMALLEST NUMBER	
+	
ARITH	Put first number in
&ARITH	SMALL
ARR1='NUMBER',RESULT='SMALL',	
&END	
ARITH	Store numbers in a
&ARITH	working array
ARR1=NUMBER',RESULT='WORK',	(WORK)
&END	
LOOP TO 100,9	
ARITH	SHIFT WORK array
&ARITH	up one number
ARR1='WORK',LOC1=2,	
RESULT='WORK',LENGTH=9,LOCR=1,	
&END	
IF (WORK(1).GE.SMALL) GO TO 90	If WORK(1) is less
ARITH	than SMALL, replace
&ARITH	SMALL by value in
ARR1='WORK',RESULT='SMALL',	WORK(1)
&END	
LABEL 90	
LABEL 100	
+	Comments
+ PRINT OUT RESULTS	
+	

Automatic Sequence	Explanation
PRTDYN &PRTDY NARRS=2, ARRNAM='NUMBER','SMALL', ARRTIT='NUMBERS','SMALLEST', OUTFMT='F6.2','F6.2', &END EOF	Print out numbers and smallest number

The following points should be noted regarding this automatic sequence:

- DRIVE1 and PRFCON begin the automatic sequence.

- Next, three dynamic arrays are allocated. The NUMBER array is filled with the 10 input numbers.

- The ARITH utility is very useful in the manipulation of dynamic arrays. (Section 2.1 of the Software Resources document provides a complete description of ARITH.) ARITH was primarily designed to perform arithmetic on dynamic arrays, but it can also be used to transfer data from one dynamic array to another. The first ARITH in the automatic sequence takes the first element from the NUMBER array, adds zero to it, and puts the result in the SMALL dynamic variable. The result is therefore a transfer of the first element of the NUMBER array into SMALL. The second ARITH uses the same technique to copy the NUMBER array into the WORK array.

- Next, the LOOP card causes the following procedure to be executed nine times:

  - Shift the WORK array up one number.
  - If WORK(1) $\geq$ SMALL, go to the last step.
  - Place the value from WORK(1) into SMALL.
  - Go back to the first step.

- Finally, the NUMBER and SMALL dynamic arrays are printed using the PRTDYN utility. (Section 5.3.10 and

Table D-6 of the User's Guide provide a complete description of the PRTDYN utility.)

OUTPUT: Figure 4-2 shows the printed output resulting from the Problem 6 run.

4.2.2 PROBLEM 7

Problem 7 is a mission analysis problem that involves trying to circularize a drag-perturbed orbit while maintaining perigee height (and therefore avoiding impact). Figure 4-3 illustrates the logical flow of this problem, which is discussed below.

PROBLEM: Given the following, and starting with a drag-perturbed satellite orbit with a perigee height of 150 kilometers and an apogee height of 1500 kilometers, propagate this orbit until the apogee height drops below 400 kilometers, keeping the perigee height between 145 and 150 kilometers.

Epoch:	January 10, 1981, 00:00:00 GMT
Elements:	Keplerian: a = 7203 kilometers e = 0.093704 i = 10 degrees Ω = 0 degrees ω = 0 degrees T = 0 degrees
Coordinate system of input:	Earth-centered true Earth equator and equinox of date
Propagator:	Time-regularized Cowell (step size = 100 steps per revolution)
Force model:	4-by-4 Earth field, effects of Moon and Sun included, Harris-Priester drag model

Print out the Keplerian elements and the osculating perifocal radius and height every 15th perigee passage. Stop when the perigee height is less than 145 kilometers. Then raise the perigee height 5 kilometers. Continue this process until the apogee height drops below 400 kilometers.

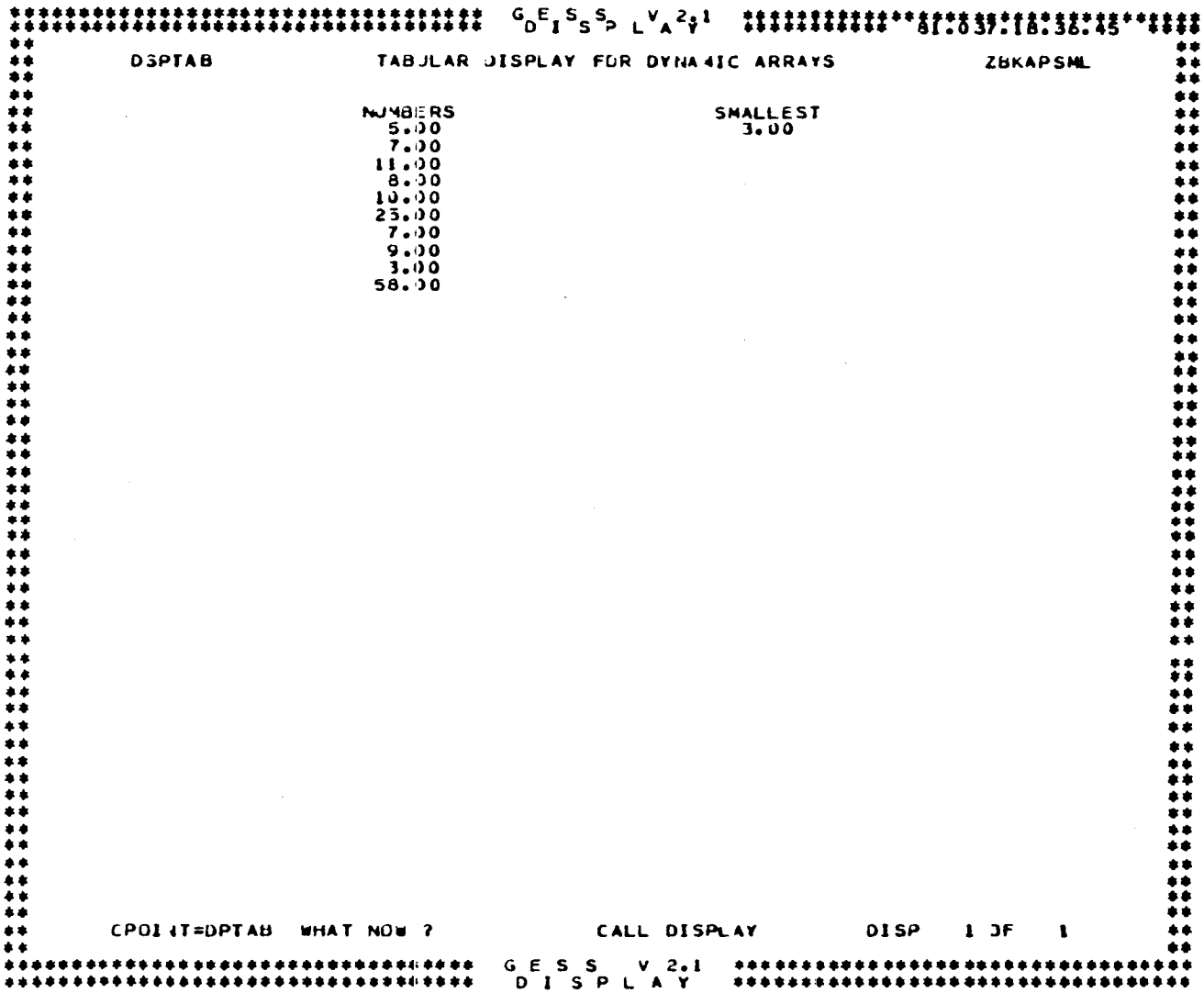


Figure 4-2. Printed Output From Problem 6 Run

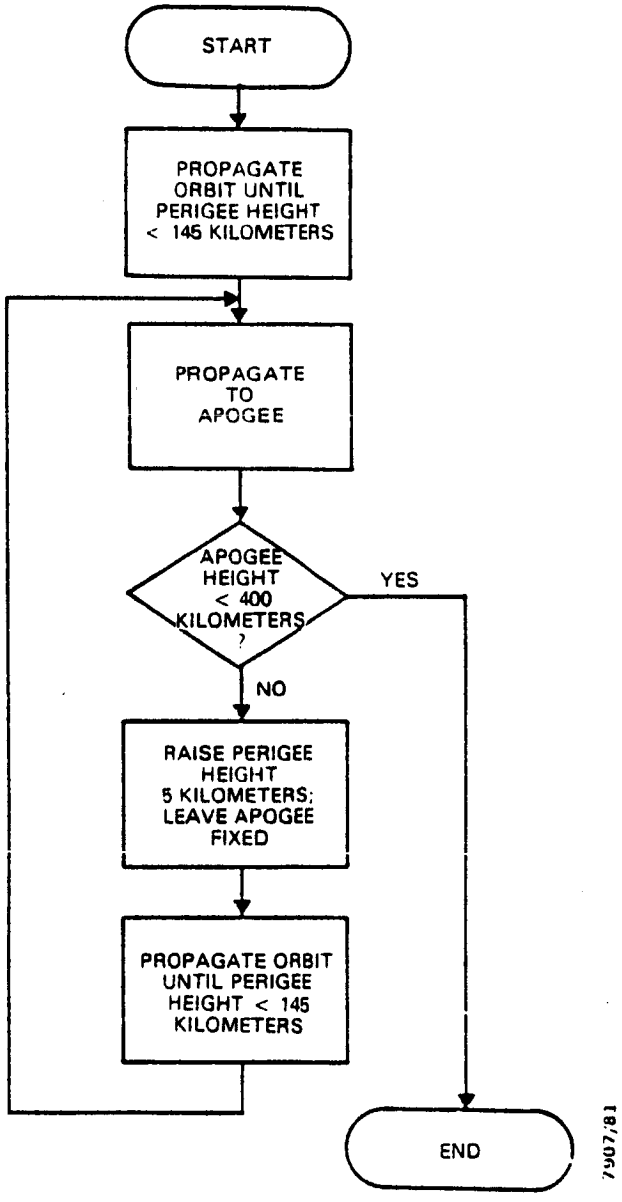


Figure 4-3. Logical Flow for Problem 7

AUTOMATIC SEQUENCE: The automatic sequence for solving this problem is presented below. It is to be noted that spaces in the sequence column are used to clarify the association of sequence entries and explanatory comments and do not indicate a break in the sequence or the use of blank cards.

Automatic Sequence	Explanation
DRIVE1	
PRFCON	
*PARMS,A,R8,2,2,2	
DYNUPD	
&DYNUP	
&END	
&DYNSEC	
IDRAG1=1,	Turn drag on
&END	
ORBINP	Process orbit input information
&ORBIN	
IELEM=2,	
ELEM=810110.,0.,	
7203.,.093704,10.,	
0.,0.,0.,	
PROPM='TRCOWL',H=100.,	
ISTOP=35,13,	
STPVAL=145.,	
IREPT=1,1,	
ISKIP=0,14,	
MULTI=1,17,	
ITERM=1,	
IPARMS=1,2,3,4,5,6,29,22,	
NOUT=4,IDYN=2,	
&END	
LABEL A	
ORBIT	Propagate until perigee height is less than 145 kilometers
DSTORE	Transfer final elements to STATEI for processing by ORBINP
&DSTOR	
XNAMEI='STATEF','STATEF',LOCI=1,3,	
XNAMEO='STATEI','STATEI',LOCO=1,3,	
NUMWDS=1,7,	
&END	
ORBINP	
&ORBIN	
IELEM=0	
PROPM='TRCOWL',H=100.,	
ISTOP=14,	

Automatic Sequence	Explanation
IPARMS=29, NOUT=5,IDYN=2 &END	
ORBIT	Propagate to apogee. Store apogee height in PARMS array
IF (PARMS(4).LE.400.0) GO TO C	If apogee height is less than 400 kilometers, exit
DELTAV &DELTA ICORDG=2, ITYPE=3, IGOAL(4)=1,IGOAL(5)=1,IGOAL(8)=1, GVAL(8)=5.0, &END	Raise perigee height 5 kilometers and place new elements in STATEI array for processing by ORBINP
ORBINP &ORBIN IELEM=0, PROPM='TRCOWL',H=100., ISTOP=35,13, STPVAL=145., IREPT=1,1 ISKIP=0,14, MULTI=1,17, ITERM=1, IPARMS=1,2,3,4,5,6,29,22, NOUT=4,IDYN=2, &END	Process orbit input information
GO TO A LABEL C EOF	

The following points should be noted concerning this automatic sequence:

- First, the PARMS array is allocated. This array is used to store the value of the current apogee height. If the current apogee height is less than 400 kilometers, GMAS execution terminates.

- DYNUPD is then called to set up the force model to be used for propagations (default plus Harris-Priester drag model).

- ORBINP is then called to provide the orbital information for the propagator. IELEM=2 informs the propagator that the input elements are Keplerian. ELEM gives the initial state elements of the satellite. The state is to be propagated using a time-regularized Cowell propagator with 100 steps per revolution (PROPM='TRCOWL',H=100.). The stopping conditions inform the propagator to stop every 15 perigee passages of the satellite until the geodetic altitude drops to 145 kilometers. At each stop the Keplerian elements, the geodetic height, and the osculating apogee height will be printed out (IPARMS). IDYN is set to 2 to include the updated dynamics file.

- ORBIT is called to propagate the satellite until the perigee height (geodetic height) drops below 145 kilometers.

- DSTORE is then called to transfer the final orbital elements from the STATEF array to the STATEI array. This must be done in order to continue propagation.

- ORBINP is called to provide orbital information for the next propagation. IELEM=0 informs the propagator that the input element state will be coming from the STATEI array instead of variable ELEM. The state will be propagated to apogee (ISTOP=14). Then the geodetic height (apogee height) at this point will be placed in the PARMS array (at position PARMS(4)). IDYN is set to 2 to include the updated dynamics file.

- ORBIT is then called to propagate the satellite from perigee (STATEI input state) to apogee and place the geodetic height in the PARMS array.

- Next, a test is performed on the apogee height. If the geodetic apogee height is less than or equal to 400 kilometers, execution continues at LABELC, and execution of the

automatic sequence terminates. Otherwise, processing continues with a call to DELTAV.

- DELTAV is called to raise the perigee height 5 kilometers and place the new orbital elements in the STATEI array for processing by ORBINP. DELTAV is a GMAS utility that is very useful for satellite maneuver approximations. In this case, it takes the current satellite state from the STATEF array. It then determines the change in the velocity vector necessary to raise perigee 5 kilometers while maintaining the other state elements. Then it adds this velocity vector to the velocity component of the STATEF vector and puts the result in the STATEI array. It also prints out a page of information relating to the change in elements. Section 2.2 of the GMAS Software Resources document provides a complete description of the DELTAV utility.

- Next, ORBINP is called to process the orbit input information. IELEM=0 informs the propagator that the input state will be through dynamic array STATEI (which was output from DELTAV in this case). The other variables under NAMELIST ORBIN are the same as those in the previous ORBIN NAMELIST.

- GO TO A causes execution to continue at label A. ORBIT is called again to propagate the satellite until the perigee height drops below 145 kilometers.

- This loop continues until apogee is less than or equal to 400 kilometers.

OUTPUT: Figure 4-4 shows the printed output resulting from the Problem 7 run.

NOTE: In pursuing this problem further, the user may want to graph the perigee and apogee height values. This can be done by allocating the proper arrays and using the PREPLT

and PLTDYN utilities. Figure 4-5 shows an automatic sequence that will accomplish this. Figure 4-6 illustrates the graphs generated by the automatic sequence in Figure 4-5.

```
DATE= 810110   0  0   0.000  (JUL|AN=2444614.50000  )   TIME FROM EPOCH=    0 DAYS  0 HRS  0 MIN   0.000SEC     TI4:
CENTRAL BODY= EARTH          COORDINATE SYSTEM= TRUE EQUATE              LOCAL TIME= 17 HRS  0 MIN   0.000 SEC
  SMA  7203.000000        ECC  .93704000000D-01   INC  10.00000000       RAN  .14923411390D-10   AP  360.0000000      TA  .14448119730D-10
MGT  149.6110680        APH  1499.869912
*****************************************************************************************************************

DATE= 810110   1 41  12.869  (JUL|AN=2444614.57029  )   TIME FROM EPOCH=    0 DAYS  1 HRS 41 MIN  12.869SEC     PERIFOCAL EARTH
CENTRAL BODY= EARTH          COORDINATE SYSTEM= TRUE EQUATE              LOCAL TIME= 18 HRS 41 MIN  12.869 SEC
  SMA  7201.749152        ECC  .93559892450-01   INC  9.994407739       RAN  359.5384744      AP  .9100186852      TA  359.9999964
MGT  149.8144389        APH  1497.404028
*****************************************************************************************************************

DATE= 810111   1 15  24.114  (JUL|AN=2444615.55236  )   TIME FROM EPOCH=    1 DAYS  1 HRS 15 MIN  24.114SEC     PERIFOCAL EARTH
CENTRAL BODY= EARTH          COORDINATE SYSTEM= TRUE EQUATE              LOCAL TIME= 17 HRS 15 MIN  24.114 SEC
  SMA  7183.658902        ECC  .91399939060-01   INC  9.996715836       RAN  353.8765378      AP  13.57298268      TA  .13121452570-04
MGT  149.2411628        APH  1462.432388
*****************************************************************************************************************

DATE= 810112   2 24  43.379  (JUL|AN=2444616.60052  )   TIME FROM EPOCH=    2 DAYS  2 HRS 24 MIN  43.379SEC     PERIFOCAL EARTH
CENTRAL BODY= EARTH          COORDINATE SYSTEM= TRUE EQUATE              LOCAL TIME= 17 HRS 24 MIN  43.379 SEC
  SMA  7164.294934        ECC  .89005341760-01   INC  9.992309501       RAN  346.1148440      AP  27.20802838      TA  .13452828390-04
MGT  148.6256709        APH  1423.815453
*****************************************************************************************************************

DATE= 810113   3 27  47.637  (JUL|AN=2444617.64430  )   TIME FROM EPOCH=    3 DAYS  3 HRS 27 MIN  47.637SEC     PERIFOCAL EARTH
CENTRAL BODY= EARTH          COORDINATE SYSTEM= TRUE EQUATE              LOCAL TIME= 17 HRS 27 MIN  47.637 SEC
  SMA  7144.030898        ECC  .86519612900-01   INC  9.987286789       RAN  339.1111190      AP  40.98913783      TA  359.9999983
MGT  148.0852644        APH  1383.989685
*****************************************************************************************************************

DATE= 810114   4 24  26.974  (JUL|AN=2444618.68365  )   TIME FROM EPOCH=    4 DAYS  4 HRS 24 MIN  26.974SEC     PERIFOCAL EARTH
CENTRAL BODY= EARTH          COORDINATE SYSTEM= TRUE EQUATE              LOCAL TIME= 18 HRS 24 MIN  26.974 SEC
  SMA  7123.215547        ECC  .83944708070-01   INC  9.982381066       RAN  332.0627108      AP  54.67908525      TA  359.9959989
MGT  147.5451503        APH  1343.031767
*****************************************************************************************************************

DATE= 810115   5 14  31.814  (JUL|AN=2444619.71442  )   TIME FROM EPOCH=    5 DAYS  5 HRS 14 MIN  31.814SEC     PERIFOCAL EARTH
CENTRAL BODY= EARTH          COORDINATE SYSTEM= TRUE EQUATE              LOCAL TIME= 18 HRS 14 MIN  31.814 SEC
  SMA  7101.875318        ECC  .81279878500-01   INC  9.978415880       RAN  324.9690877      AP  68.52011805      TA  .13793230625D-05
MGT  147.0544047        APH  1360.974881
*****************************************************************************************************************

DATE= 810116   6 37  51.712  (JUL|AN=2444620.74852  )   TIME FROM EPOCH=    6 DAYS  6 HRS 37 MIN  51.712SEC     PERIFOCAL EARTH
CENTRAL BODY= EARTH          COORDINATE SYSTEM= TRUE EQUATE              LOCAL TIME= 18 HRS 37 MIN  51.712 SEC
  SMA  7079.693017        ECC  .78519622350-01   INC  9.976085989       RAN  317.8318694      AP  82.43276836      TA  .13652921980-05
MGT  146.5682728        APH  1257.771423
*****************************************************************************************************************

DATE= 810117   6 34  15.809  (JUL|AN=2444621.77379  )   TIME FROM EPOCH=    7 DAYS  6 HRS 34 MIN  15.809SEC     PERIFOCAL EARTH
CENTRAL BODY= EARTH          COORDINATE SYSTEM= TRUE EQUATE              LOCAL TIME= 19 HRS 34 MIN  15.809 SEC
  SMA  7057.557419        ECC  .75660369100-01   INC  9.975767985       RAN  310.6542351      AP  96.41756467      TA  .13776979080-04
MGT  146.0772303        APH  1213.395386
*****************************************************************************************************************

DATE= 810118   7  3  32.233  (JUL|AN=2444622.79412  )   TIME FROM EPOCH=    8 DAYS  7 HRS  3 MIN  32.233SEC     PERIFOCAL EARTH
CENTRAL BODY= EARTH          COORDINATE SYSTEM= TRUE EQUATE              LOCAL TIME= 20 HRS  3 MIN  32.233 SEC
  SMA  7034.489359        ECC  .72690753380-01   INC  9.977414260       RAN  303.4397781      AP  110.4749339      TA  .13792334830-04
MGT  145.5731294        APH  1167.691690
*****************************************************************************************************************

DATE= 810119   7 25  27.566  (JUL|AN=2444623.80925  )   TIME FROM EPOCH=    9 DAYS  7 HRS 25 MIN  27.566SEC     PERIFOCAL EARTH
CENTRAL BODY= EARTH          COORDINATE SYSTEM= TRUE EQUATE              LOCAL TIME= 20 HRS 25 MIN  27.566 SEC
  SMA  7013.673816        ECC  .69598023980-01   INC  9.980550452       RAN  296.1912094      AP  124.6074928      TA  359.9999924
MGT  145.0567600        APH  1120.448516
*****************************************************************************************************************

DATE= 810120   2 49  27.860  (JUL|AN=2444624.61768  )   TIME FROM EPOCH=   10 DAYS  2 HRS 49 MIN  27.860SEC     ALTITUDE
CENTRAL BODY= EARTH          COORDINATE SYSTEM= TRUE EQUATE              LOCAL TIME= 20 HRS 49 MIN  27.860 SEC
  SMA  6990.686338        ECC  .66987974990-01   INC  9.983211346       RAN  290.3774772      AP  135.9292399      TA  359.7483796
MGT  144.8535349        APH  1081.158356
*****************************************************************************************************************
```

Figure 4-4. Printed Output From Problem 7 Run (1 of 9)

MANEUVER SUMMARY

IGNITION DATE(YYMMDD) 810120.0
TIME(HHMMSS) 33755.2
END DATE(YYMMDD) 810120.0
TIME(HHMMSS) 33755.2
BURN ESTIMATE(SEC) 0.0

VARIABLE(UNITS)	IGNITION	END THRUST	NO THRUST	DELTA(END-NO)
1)CSCULATING(MEAN 1950) KEPLERIAN				
SMA(KM)	6987.679	6990.179	6987.679	2.500
ECC(N/A)	0.0642805	0.0638999	0.0642805	-0.0003806
INC(DEG)	9.823324	9.823024	9.823324	0.000000
RA NODE(DEG)	290.09459	290.09456	290.09459	0.0
ARG PER(DEG)	136.05806	136.05807	136.05806	0.00001
MA(DEG)	180.00000	179.99996	180.00000	-0.00001
CARTESIAN				
X(KM)	-2935.676	-2935.676	-2935.676	-0.000
Y(KM)	-6775.642	-6775.642	-6775.642	-0.000
Z(KM)	-880.435	-880.435	-880.435	-0.000
VX(KM/SEC)	6.407046	6.408346	6.407046	0.001303
VY(KM/SEC)	-2.888890	-2.889478	-2.888890	-0.000588
VZ(KM/SEC)	0.869959	0.870136	0.869959	0.000177
2)RTN VELOCITY VECTOR				
VR(KM/SEC)	-0.000000	0.000000	-0.000000	0.000000
VT(KM/SEC)	7.081861	7.083301	7.081861	0.001440
VN(KM/SEC)	0.000000	0.000000	0.000000	0.000000
3)MISCELLANEOUS ORBIT				
RMAG(KM)	7436.851	7436.851	7436.851	0.000
VMAG(KM/SEC)	7.081861	7.083301	7.081861	0.001440
TA(DEG)	180.00000	179.99996	180.00000	-0.00001
LAT(DEG+N)	-6.799084	-6.799084	-6.799084	-0.000000
LONG(DEG+E)	72.74288	72.74288	72.74288	-0.00000
FPA(DEG)	90.000000	90.000000	90.000000	-0.000001
DRIFT(DEG/DAY(E)	335.71239	335.69927	335.71230	-0.01304
GHA(DEG)	3.03395	3.03395	3.03395	0.0
PERIOD(HR)	1.614759	1.615626	1.614759	0.000867
APOGEE(KM)	7436.851	7436.851	7436.851	0.000
PERIGEE(KM)	6538.507	6543.507	6538.507	5.000
PLANE CHG(DEG)	0.0	0.0	0.0	0.0
DV(KM/SEC)	0.0	0.0014402	0.0	0.0014402
FUEL(LB)	0.0	0.0	0.0	0.0
4)TRUE EARTH EQUATOR AND EQUINOX OF DATE KEPLERIAN				
SMA(KM)	6987.679	6990.179	6987.679	2.500
ECC(N/A)	0.0642805	0.0638999	0.0642805	-0.0003806
INC(DEG)	9.983513	9.983513	9.983513	0.000000
RA NODE(DEG)	290.14166	290.14166	290.14166	0.0
ARG PER(DEG)	136.41094	136.41094	136.41094	0.00001
MA(DEG)	180.00000	179.99996	180.00000	-0.00001
CARTESIAN				
X(KM)	-2886.280	-2886.280	-2886.280	-0.000
Y(KM)	-6796.022	-6796.022	-6796.022	-0.000
Z(KM)	-888.940	-888.940	-888.940	-0.000
VX(KM/SEC)	6.424160	6.425467	6.424160	0.001306
VY(KM/SEC)	-2.844668	-2.845246	-2.844668	-0.000579
VZ(KM/SEC)	0.889260	0.889441	0.889260	0.000181

Figure 4-4. Printed Output From Problem 7 Run (2 of 9)

```
                                    GMAS PARAMETER REPORT                                    PAGE    1

DATE= 810120    3 37  55.161  (JUL DAN=2444623.65133    )  TIME FROM EPOCH=    0 DAYS  8 HRS 37 MIN  55.161 SEC      TIME
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE      LOCAL TIME= 8 HRS 37 MIN  55.161 SEC
  SMA  6967.181385       ECC  .6389954317D-01  INC  9.983512674       RAN  290.1416614       AP  136.4109446       TA  179.9999940
MGT  1059.017283       APH  1058.718503
*****************************************************************************************************************

DATE= 810120    4 26  20.043  (JUL DAN=2444624.68425    )  TIME FROM EPOCH=    0 DAYS  0 HRS 48 MIN  24.882SEC      PERIFOCAL EARTH
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE      LOCAL TIME= 20 HRS 26 MIN  20.043 SEC
  SMA  6971.672333       ECC  .6642289332D-01  INC  9.983338966       RAN  289.8889057       AP  136.8860070       TA  359.9999983
MGT  149.7626680       APH  1078.259366
*****************************************************************************************************************

DATE= 810121    2 59   9.023  (JUL DAN=2444625.62441    )  TIME FROM EPOCH=    0 DAYS 23 HRS 21 MIN  13.882SEC      PERIFOCAL EARTH
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE      LOCAL TIME= 20 HRS 59 MIN   9.023 SEC
  SMA  6973.643399       ECC  .6398642370D-01  INC  9.986726935       RAN  283.9775195       AP  150.1929723       TA  .1496432110D-05
MGT  149.4445850       APH  1041.721900
*****************************************************************************************************************

DATE= 810122    3  2  33.651  (JUL DAN=2444626.62078    )  TIME FROM EPOCH=    1 DAYS 23 HRS 24 MIN  36.490SEC      PERIFOCAL EARTH
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE      LOCAL TIME= 21 HRS  2 MIN  33.651 SEC
  SMA  6953.194319       ECC  .6129595821D-01  INC  9.989314608       RAN  275.7500091       AP  164.5295641       TA  359.9999936
MGT  143.3916175       APH  1001.463029
*****************************************************************************************************************

DATE= 810123    2 59  33.121  (JUL DAN=2444627.62469    )  TIME FROM EPOCH=    2 DAYS 23 HRS 21 MIN  36.163SEC      PERIFOCAL EARTH
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE      LOCAL TIME= 21 HRS 59 MIN  33.121 SEC
  SMA  6932.466338       ECC  .5849975481D-01  INC  9.990384893       RAN  268.3892924       AP  178.9509932       TA  .3376518738D-06
MGT  138.7729736       APH  959.8739187
*****************************************************************************************************************

DATE= 810124    2 49  56.935  (JUL DAN=2444628.61902    )  TIME FROM EPOCH=    3 DAYS 23 HRS 12 MIN   1.774SEC      PERIFOCAL EARTH
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE      LOCAL TIME= 21 HRS 49 MIN  56.935 SEC
  SMA  6910.794158       ECC  .5559440652D-01  INC  9.989648254       RAN  260.9887165       AP  193.4671091       TA  .3489193060D-06
MGT  138.4877281       APH  916.8556601
*****************************************************************************************************************

DATE= 810125    2 33  32.222  (JUL DAN=2444629.60682    )  TIME FROM EPOCH=    4 DAYS 22 HRS 55 MIN  37.081SEC      PERIFOCAL EARTH
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE      LOCAL TIME= 22 HRS 33 MIN  32.222 SEC
  SMA  6888.295605       ECC  .5257049910D-01  INC  9.987257701       RAN  253.5411858       AP  208.0869269       TA  .3743327544D-06
MGT  138.1776779       APH  872.2766033
*****************************************************************************************************************

DATE= 810126    2 17   4.706  (JUL DAN=2444630.59033    )  TIME FROM EPOCH=    5 DAYS 22 HRS 32 MIN   9.565SEC      PERIFOCAL EARTH
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE      LOCAL TIME= 23 HRS 10 MIN   4.706 SEC
  SMA  6864.633545       ECC  .4941208175D-01  INC  9.985784227       RAN  246.0403340       AP  222.8182395       TA  .3931342410D-06
MGT  137.7821982       APH  825.8992612
*****************************************************************************************************************

DATE= 810127    1 39  15.909  (JUL DAN=2444631.56893    )  TIME FROM EPOCH=    6 DAYS 22 HRS  1 MIN  20.748SEC      PERIFOCAL EARTH
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE      LOCAL TIME= 23 HRS 39 MIN  15.909 SEC
  SMA  6840.171573       ECC  .4648948844D-01  INC  9.984391861       RAN  238.4812173       AP -122.3327347       TA  359.9999998
MGT  137.2314068       APH  777.2906105
*****************************************************************************************************************

DATE= 810128    1 15  40.760  (JUL DAN=2444632.54214    )  TIME FROM EPOCH=    7 DAYS 21 HRS 22 MIN  45.599SEC      PERIFOCAL EARTH
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE      LOCAL TIME=  0 HRS  0 MIN  40.760 SEC
  SMA  6813.508937       ECC  .4255137694D-01  INC  9.977129248       RAN  230.8610291       AP  252.6390380       TA  .3913588992D-06
MGT  136.4152879       APH  725.7101449
*****************************************************************************************************************

DATE= 810129    1 13  43.527  (JUL DAN=2444633.50653    )  TIME FROM EPOCH=    8 DAYS 20 HRS 35 MIN  48.306SEC      PERIFOCAL EARTH
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE      LOCAL TIME=  0 HRS 13 MIN  43.527 SEC
  SMA  6785.343239       ECC  .3870511326D-01  INC  9.975669601       RAN  223.1784210       AP -92.25947022       TA  359.9999938
MGT  135.2155492       APH  669.8307172
*****************************************************************************************************************

DATE= 810130    1 18  38.410  (JUL DAN=2444634.53781    )  TIME FROM EPOCH=    9 DAYS 23 HRS 40 MIN  40.249SEC      ALL/TRUE
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE      LOCAL TIME=  0 HRS 18 MIN  38.410 SEC
  SMA  6781.381338       ECC  .3819000615D-01  INC  9.976209355       RAN  222.1476163       AP -90.30615410       TA  359.2297259
MGT  134.9266570       APH  662.2223325
*****************************************************************************************************************
```

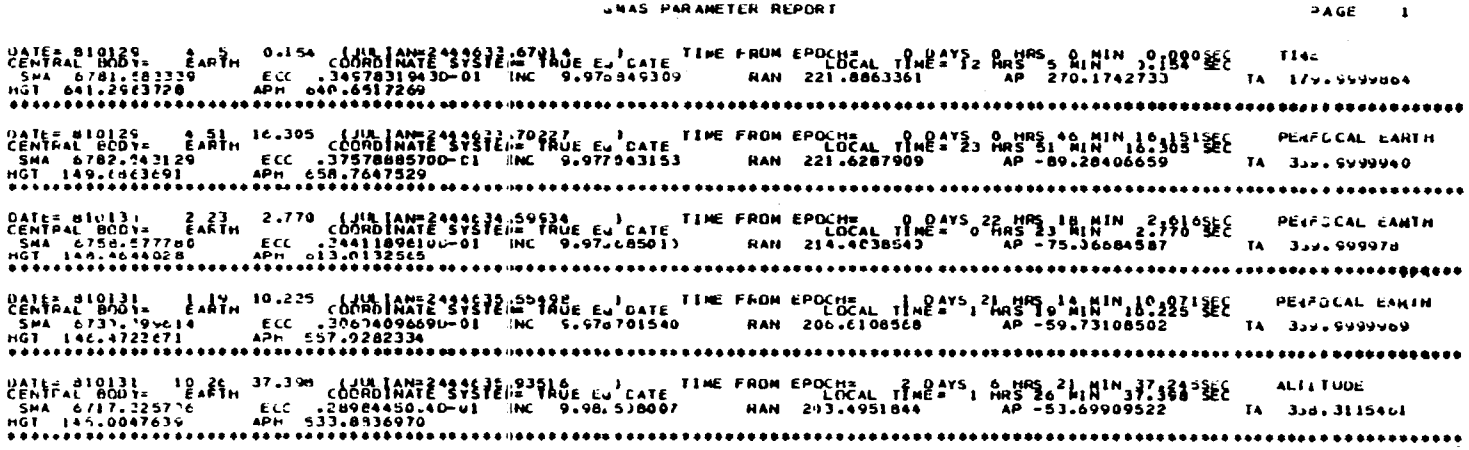
Figure 4-4. Printed Output From Problem 7 Run (3 of 9)

MANEUVER SUMMARY

IGNITION DATE(YYMMDD) 810129.
TIME(HHMMSS) 40500.2
END DATE(YYMMDD) 810129.0
TIME(HHMMSS) 40500.2
BURN ESTIMATE(SEC) 0.0

VARIABLE(UNITS)	IGNITION	END THRUST	NC THRUST	DELTA(END-NO)
1)OSCULATING(MEAN 1950) KEPLERIAN				
SMA(KM)	6776.381	6781.581	6779.381	2.500
ECC(N/A)	0.0353604	0.0346787	0.0353604	-0.0003817
INC(DEG)	9.862206	9.862206	9.862206	0.000000
RA NODE(DEG)	220.76362	220.76662	220.76362	-0.00000
ARG PER(DEG)	270.91613	270.91615	270.91613	0.00002
MA(DEG)	180.00001	179.99995	180.00001	-0.00002
CARTESIAN				
X(KM)	4599.276	4599.276	4599.276	-0.000
Y(KM)	-5163.840	-5163.840	-5163.840	-0.000
Z(KM)	1202.020	1202.020	1202.020	-0.000
VX(KM/SEC)	5.529391	5.530484	5.529391	0.001094
VY(KM/SEC)	4.920143	4.921116	4.920143	0.000973
VZ(KM/SEC)	-0.020270	-0.020274	-0.020270	-0.000004
2)RTN VELOCITY VECTOR				
VR(KM/SEC)	-0.000000	0.000000	-0.000000	0.000000
VT(KM/SEC)	7.401512	7.402976	7.401512	0.001464
VN(KM/SEC)	-0.000000	-0.000000	-0.000000	-0.000000
3)MISCELLANEOUS ORBIT				
RMAG(KM)	7018.792	7018.792	7018.792	-0.000
VMAG(KM/SEC)	7.4015121	7.4029762	7.4015121	0.0014641
TA(DEG)	180.00001	179.99995	180.00001	-0.00002
LAT(DEG+N)	9.860933	9.860933	9.860933	0.000000
LONG(DEG+E)	122.19801	122.19801	122.19801	-0.00000
VPA(DEG)	90.000000	90.000000	90.000000	-0.000001
DRIFT(DEG/DAY+E)	336.79171	336.77887	336.79171	-0.01284
GHA(DEG)	3.30727	3.30727	3.30727	0.0
PERIOD(HR)	1.542995	1.543849	1.542995	0.000854
APOGEE(KM)	7018.792	7018.792	7018.792	0.000
PERIGEE(KM)	6539.370	6544.370	6539.370	5.000
PLANE CHG(DEG)	0.0	0.0	0.0	0.0
DV(KM/SEC)	0.0	0.0014641	0.0	0.0014641
FUEL(LB)	0.0	0.0	0.0	0.0
4)TRUE EARTH EQUATOR AND EQUINOX OF DATE KEPLERIAN				
SMA(KM)	6776.381	6781.581	6779.381	2.500
ECC(N/A)	0.0353604	0.0346787	0.0353604	-0.0003817
INC(DEG)	9.976849	9.976849	9.976849	0.000000
RA NODE(DEG)	221.88634	221.88634	221.88634	-0.00000
ARG PER(DEG)	270.17425	270.17427	270.17425	0.00002
MA(DEG)	180.00001	179.99995	180.00001	-0.00002
CARTESIAN				
X(KM)	4631.138	4631.138	4631.138	-0.000
Y(KM)	-5131.992	-5131.992	-5131.992	-0.000
Z(KM)	1216.002	1216.002	1216.002	-0.000
VX(KM/SEC)	5.495382	5.496469	5.495382	0.001087
VY(KM/SEC)	4.958139	4.959120	4.958139	0.000981
VZ(KM/SEC)	-0.003900	-0.003901	-0.003900	-0.000001

Figure 4-4. Printed Output From Problem 7 Run (4 of 9)



```
                                    MAS PARAMETER REPORT                                    PAGE    1

DATE= 810129    4 5    0.154  (JUL|AN=2444633.67014    )  TIME FROM EPOCH=    0 DAYS  0 HRS  0 MIN   0.000SEC     TIME
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE          LOCAL TIME= 12 HRS  5 MIN   0.154 SEC
  SMA   6781.583339       ECC  .3467831943D-01   INC   9.976849309       RAN  221.8863361       AP  270.1742733       TA  179.9999984
MGT  641.2963728        APH  640.6517269
*****************************************************************************************************************

DATE= 810129    4 51  16.305  (JUL|AN=2444633.70227    )  TIME FROM EPOCH=    0 DAYS  0 HRS 46 MIN  16.151SEC     PERIFOCAL EARTH
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE          LOCAL TIME= 23 HRS 51 MIN  16.305 SEC
  SMA   6782.243129       ECC  .3757888570D-01   INC   9.977943153       RAN  221.6287909       AP  -89.28406659       TA  359.9999960
MGT  149.0663691        APH  658.7647529
*****************************************************************************************************************

DATE= 810131    2 23   2.770  (JUL|AN=2444634.59634    )  TIME FROM EPOCH=    0 DAYS 22 HRS 18 MIN   2.616SEC     PERIFOCAL EARTH
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE          LOCAL TIME=  0 HRS 23 MIN   2.770 SEC
  SMA   6758.577780       ECC  .2441189810D-01   INC   9.972685011       RAN  214.4038543       AP  -75.36684587       TA  359.9999978
MGT  144.4644028        APH  613.0132565
*****************************************************************************************************************

DATE= 810131    1 19  10.225  (JUL|AN=2444635.55498    )  TIME FROM EPOCH=    1 DAYS 21 HRS 14 MIN  10.071SEC     PERIFOCAL EARTH
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE          LOCAL TIME=  1 HRS 19 MIN  10.225 SEC
  SMA   6731.194614       ECC  .3067409669D-01   INC   9.970701540       RAN  206.6108568       AP  -59.73108502       TA  359.9999969
MGT  146.4722671        APH  557.9282334
*****************************************************************************************************************

DATE= 810131   10 26  37.398  (JUL|AN=2444635.93516    )  TIME FROM EPOCH=    2 DAYS  6 HRS 21 MIN  37.243SEC     ALTITUDE
CENTRAL BODY=   EARTH          COORDINATE SYSTEM= TRUE EQ DATE          LOCAL TIME=  1 HRS 26 MIN  37.398 SEC
  SMA   6717.225706       ECC  .2898445D-01   INC   9.985D8007       RAN  203.4951844       AP  -53.69909522       TA  358.3115401
MGT  145.0047639        APH  533.8936970
*****************************************************************************************************************
```

Figure 4-4. Printed Output From Problem 7 Run (5 of 9)

MANEUVER SUMMARY

IGNITION DATE(YYMMDD) 810131.0
TIME(HHMMSS) 111237.1
END DATE(YYMMDD) 810131.0
TIME(HHMMSS) 111237.1
BURN ESTIMATE(SEC) 0.0

VARIABLE(UNITS)	IGNITION	END THRUST	NO THRUST	DELTA(END-NO)
1)OSCULATING(MEAN 1950) KEPLERIAN				
SMA(KM)	6715.104	6717.604	6715.104	2.500
ECC(N/A)	0.0260333	0.0256515	0.0260333	-0.0003818
INC(DEG)	9.913868	9.913868	9.913868	0.000000
RA NODE(DEG)	201.94102	201.94102	201.94102	-0.00000
ARG PER(DEG)	307.77694	307.77696	307.77654	0.00031
MA(DEG)	180.00002	180.00030	180.00032	-0.00002
CARTESIAN				
X(KM)	5919.432	5919.432	5919.432	-0.000
Y(KM)	-3398.869	-3398.868	-3398.868	0.000
Z(KM)	937.590	937.590	937.589	-0.000
VX(KM/SEC)	3.810829	3.811576	3.810829	0.000747
VY(KM/SEC)	6.418525	6.419783	6.418525	0.001258
VZ(KM/SEC)	-0.791689	-0.791844	-0.791689	-0.000155
2)RTN VELOCITY VECTOR				
VR(KM/SEC)	-0.000000	-0.000000	-0.000000	0.000000
VT(KM/SEC)	7.506441	7.507912	7.506441	0.001471
VN(KM/SEC)	-0.000000	-0.000000	-0.000000	-0.000000
3)MISCELLANEOUS ORBIT				
RMAG(KM)	6889.920	6889.920	6889.920	-0.000
VMAG(KM/SEC)	7.5064405	7.5079118	7.5064405	0.0014713
TA(DEG)	180.00002	180.00000	180.00002	-0.00001
LAT(DEG+N)	7.821160	7.821160	7.821160	-0.000000
LONG(DEG+E)	31.47550	31.47556	31.47550	0.0
VPA(DEG)	90.000000	90.000000	90.000000	-0.000000
DRIFT(DEG/DAYME)	337.11947	337.10670	337.11947	-0.01278
GMA(DEG)	5.21261	5.21261	5.21261	0.0
PERIOD(HR)	1.521204	1.522053	1.521204	0.000850
APOGEE(KM)	6889.920	6889.920	6889.920	-0.000
PERIGEE(KM)	6540.287	6545.287	6540.287	5.000
PLANE CHG(DEG)	0.0	0.0	0.0	0.0
DV(KM/SEC)	0.0	0.0014713	0.0	0.0014713
FUEL(LB)	0.0	0.0	0.0	0.0
4)TRUE EARTH EQUATOR AND EQUINOX OF DATE KEPLERIAN				
SMA(KM)	6715.104	6717.604	6715.104	2.500
ECC(N/A)	0.0260333	0.0256515	0.0260333	-0.0003818
INC(DEG)	9.981406	9.981406	9.981406	0.000000
RA NODE(DEG)	203.23628	203.23628	203.23628	-0.00000
ARG PER(DEG)	306.86297	306.86298	306.86297	0.00001
MA(DEG)	180.00002	180.00034	180.00002	-0.00032
CARTESIAN				
X(KM)	5939.888	5939.888	5939.888	-0.000
Y(KM)	-3357.949	-3357.949	-3357.949	0.000
Z(KM)	955.462	955.462	955.462	-0.000
VX(KM/SEC)	3.768842	3.769581	3.768842	0.000739
VY(KM/SEC)	6.444630	6.445893	6.444630	0.001263
VZ(KM/SEC)	-0.780523	-0.780676	-0.780523	-0.000153

Figure 4-4. Printed Output From Problem 7 Run (6 of 9)

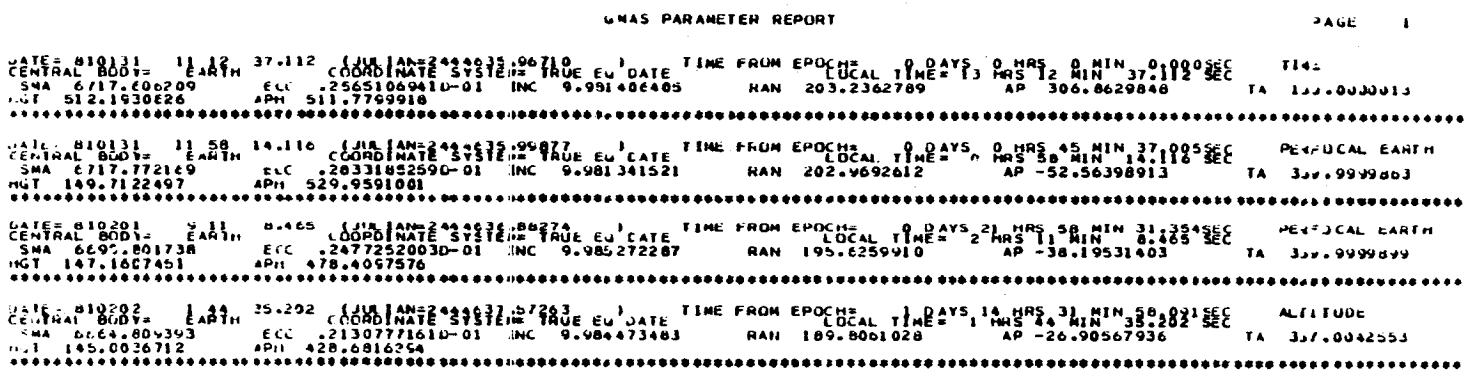


Figure 4-4. Printed Output From Problem 7 Run (7 of 9)

MANEUVER SUMMARY

IGNITION DATE(YYMMDD) 810202.0
TIME(HHMMSS) 23021.8
END DATE(YYMMDD) 810202.0
TIME(HHMMSS) 23021.8
BURN ESTIMATE(SEC) 0.0

VARIABLE(UNITS)	IGNITION	END THRUST	NO THRUST	DELTA(END-NO)
1)OSCULATING(MEAN 1950) KEPLERIAN				
SMA(KM)	6662.655	6665.155	6662.655	2.500
ECC(N/A)	0.0182073	0.0178254	0.0182073	-0.0003819
INC(DEG)	9.957192	9.957 82	9.957162	-0.000030
RA NODE(DEG)	188.17905	188.17905	188.17905	0.0
ARG PER(DEG)	334.66847	334.66850	334.66847	0.00003
MA(DEG)	18 .00001	179.99998	180.00001	-0.00003
CARTESIAN				
X(KM)	6476.016	6476.016	6476.016	0.000
Y(KM)	-1957.421	-1957.421	-1957.421	0.000
Z(KM)	501.882	501.882	501.882	-0.000
VX(KM/SEC)	2.254642	2.255081	2.254642	0.000439
VY(KM/SEC)	7.155007	7.156399	7.155007	0.001391
VZ(KM/SEC)	-1.187008	-1.187235	-1.187008	-0.000231
2)SIN VELOCITY VECTOR				
VR(KM/SEC)	-0.000000	0.000000	-0.000000	0.000000
VT(KM/SEC)	7.595165	7.596642	7.595165	0.001477
VN(KM/SEC)	-0.000000	0.0	0.0	0.0
3)MISCELLANEOUS ORBIT				
RMAG(KM)	6783.964	6783.964	6783.964	0.000
VMAG(KM/SEC)	7.5951649	7.5966421	7.5951649	0.0014771
TA(DEG)	180.00001	179.99998	180.00001	-0.00003
LAT(DEG+N)	4.242654	4.242654	4.242654	-0.000000
LONG(DEG+E)	173.47441	173.47441	173.47441	0.00000
VPA(DEG)	90.000000	90.000000	90.000000	-0.000000
DRIFT(DEG/DAY+E)	337.38701	337.37428	337.39771	-0.01273
GMA(DEG)	2.96196	2.96196	2.96196	0.0
PERIOD(MN)	1.503416	1.5034203	1.503416	0.0004846
APOGEE(KM)	6783.964	6783.964	6783.964	-0.000
PERIGEE(KM)	6541.346	6546.346	6541.346	5.000
PLANE CHG(DEG)	0.0	0.0	0.0	0.0
DV(KM/SEC)	0.0	0.0014771	0.0	0.0014771
FUEL(LB)	0.0	0.0	0.0	0.0
4)TRUE EARTH EQUATOR AND EQUINOX OF DATE KEPLERIAN				
SMA(KM)	6662.655	6665.155	6662.655	2.500
ECC(N/A)	0.0182073	0.0178254	0.0182073	-0.0003819
INC(DEG)	9.965213	9.965213	9.965213	-0.000000
RA NODE(DEG)	189.53796	189.53796	189.53796	0.00000
ARG PER(DEG)	333.68689	333.68691	333.68685	0.00003
MA(DEG)	180.00001	179.99998	180.00001	-0.00003
CARTESIAN				
X(KM)	6487.827	6487.827	6487.827	0.000
Y(KM)	-1912.706	-1912.706	-1912.706	0.000
Z(KM)	521.370	521.370	521.370	-0.000
VX(KM/SEC)	2.208797	2.209226	2.208797	0.000430
VY(KM/SEC)	7.173362	7.171757	7.170362	0.001394
VZ(KM/SEC)	-1.180530	-1.180760	-1.180530	-0.000230

Figure 4-4. Printed Output From Problem 7 Run (8 of 9)

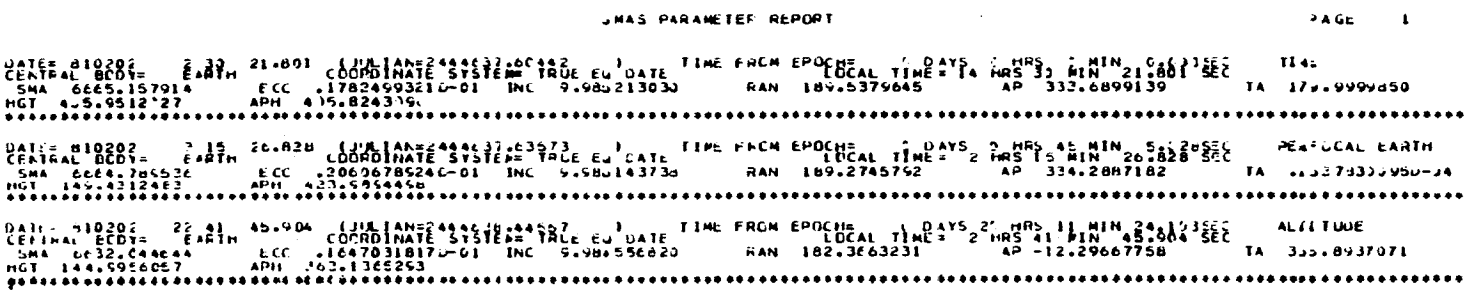


Figure 4-4. Printed Output From Problem 7 Run (9 of 9)

```
                                    USER CARD INPUT FOR THIS RUN

 &GMASEX  SEQNAM='CARDS',IUPDAT=0,IBATCH=1, &END
DRIVE1
+     ALLOCATE DYNAMIC ARRAYS
*PARMS,A,R8,2,9,20
*MAXPLT,A,I4,1,1
 &VALUE  D=1*0, &END
*EPOCH,A,R8,1,1
*MAXITR,A,I4,1,1
 &VALUE D=1*, &END
*ICOUNT,A,I4,1,1
*TIME,A,R8,1,100
*PERHT,A,R8,1,100
 &VALUE D=100*150.0, &END
*APOHT,A,R8,1,100
 &VALUE D=100*1500.0, &END
*NPLOT,A,I4,1,11
PREFCON
+     SET FORCE MODEL FOR PROPAGATION
DYNUPD
 &DYNUP
 &END
 &DYNSEC
  IDPAG1=1,
 &END
+     INPUT INITIAL STATE, STOPPING CONDITIONS
ORBINP
 &ORBIN
  IELEM=2,
  ELEM=810110.0, 0.0,
       7203.0, 0.093704, 10.0, 3*0.0,
  PROPM='TRCOWL', M=100.0,
  ISTOP=1,35,12,13,
  STPVAL=2.592D6,145.0,
  IREPT=1,1,1,
  ISKIP=0,0,14,
  MULTI=1,1,17,
  ITERM=1,2,
  IPARMS=1,2,3,4,5,6,29,22,
  NOUT=6, IDYN=2,
 &END
+     SAVE JD OF EPOCH
DSTORE
 &DSTOR
  XNAMEI='STATEI',LOCI=1,
  XNAMEO='EPOCH',LOCO=1,
  NUMWDS=1,
 &END
+     INITIALIZE TIME ARRAY SO THAT POSSIBLE
+     UNUSED POINTS WILL NOT PLOT AS ZERO
ARITH
 &ARITH
  ARR1='TIME',
  ARR2='EPOCH',
  CONST='ARR2',
  LGNTH=100,
 &END
+     BEGIN LOOP, INCREMENT AND TEST ITERATION COUNTER
LABEL A
INCREM
 &INCRE
  XNAME='ICOUNT', LOC=1, INCR=1.0,
 &END
IF (ICOUNT .GT. MAXITR) GO TO C
+     PROPAGATE UNTIL PERIGEE FALLS
ORBIT
+     TEST FOR PLOT ARRAYS FULL
IF (NPLOT(1) .GE. MAXPLT) GO TO C
+     PUT PARMS INFO IN PLOT ARRAYS
LABEL B
PREPLT,LXK
 &PREPL
  INDEX=1,8,9,
  XNAME='TIME','PERHT','APOHT',
  ISTART=2,2,2,
 &END
+     MOVE STATEF TO STATEI FOR PROPAGATOR RESTART
DSTORE
 &DSTOR
  XNAMEI='STATEF','STATEF',LOCI=1,3,
  XNAMEO='STATEI','STATEI',LOCO=1,3,
  NUMWDS=1,7,
 &END
+     SET APOGEE STOP
```

Figure 4-5. GMAS Automatic Sequence for Plot Solution to Problem 7 (1 of 2)

```
ORBINP
 &ORBIN
  IELEM='',
  PROPM='TPCOWL', H=100.0,
  ISTOP=1,14,
  ITERM=2,
  IPARMS=29,
  NOUT=5,
  IDYN=2,
 &END
+     PROPAGATE TO APOGEE FOR DELTAV
+     (MUST RAISE PERIGEE FROM APOGEE)
ORBIT
+     TEST FOR END OF STUDY
IF (PARMS(11) .LE. 400.0) GO TO C
+     RAISE PERIGEE 5.. KM
DELTAV
 &DELTA
  ICONDG=2,
  ITYPE=3,
  IGCAL(4)=1, IGCAL(5)=1,
  IGCAL(8)=1,
  GVAL(8)=5.0,
 &END
+     RESET STOPPING CONDITIONS AND RETURN FOR NEXT ITERATION
ORBINP
*PARMS,I,38,2,9,2)
 &VALUE D=180*0.0, &END
 &ORBIN
  IELEM='',
  PROPM='TPCOWL', H=100.0,
  ISTOP=1,35,13,13,
  STPVAL=2.592D6,145.,
  ITEPT=1,1,1,
  ISKIP=0,0,14,
  MULTI=1,1,17,
  ITERM=1,2,
  IPARMS=1,2,3,4,5,6,29,22,
  NOUT=5,  IDYN=2,
 &END
GO TO A
+     ADJUST TIME ARRAY TO GIVE DAYS FROM EPOCH
LABEL C
ARITH
 &ARITH
  ARR1='TIME',
  ARR2='EPOCH',
  CONST='ARR2',
  OP='-',  LGNTH=100,
 &END
+     GENERATE PLOTS
PLTDYN
 &PLTDY
  NYARPS=1,
  XARRAY='TIME',
  YARRAY='PERHT',
  IPLOT=1, IGRID=0,
  XTITLE='DAYS FROM EPOCH',
  YTITLE='GEODETIC PERIGEE HEIGHT',
 &END
PLTDYN
 &PLTDY
  NYARPS=1,
  XARRAY='TIME',
  YARRAY='APOHT',
  IPLOT=1, IGRID=0,
  XTITLE='DAYS FROM EPOCH',
  YTITLE='EQUATORIAL APOGEE HEIGHT',
 &END
+     IF PLOTS WERE GENERATED BECAUSE PLOT ARRAYS WERE FULL,
+     ZERO NPLOT ARRAY, RESTORE TIME ARRAY TO JD, AND CONTINUE
IF (NPLOT(1) .LT. MAXPLT) GO TO D
ARITH
*NPLOT,I,14,1,11
 &VALUE  D=11*0,  &END
 &ARITH
  ARR1='TIME',
  ARR2='EPOCH',
  CONST='ARR2',
  LGNTH=100,
 &END
GO TO B
+     END OF AUTOMATIC SEQUENCE
LABEL D
EOF
```

Figure 4-5. GMAS Automatic Sequence for Plot Solution to Problem 7 (2 of 2)

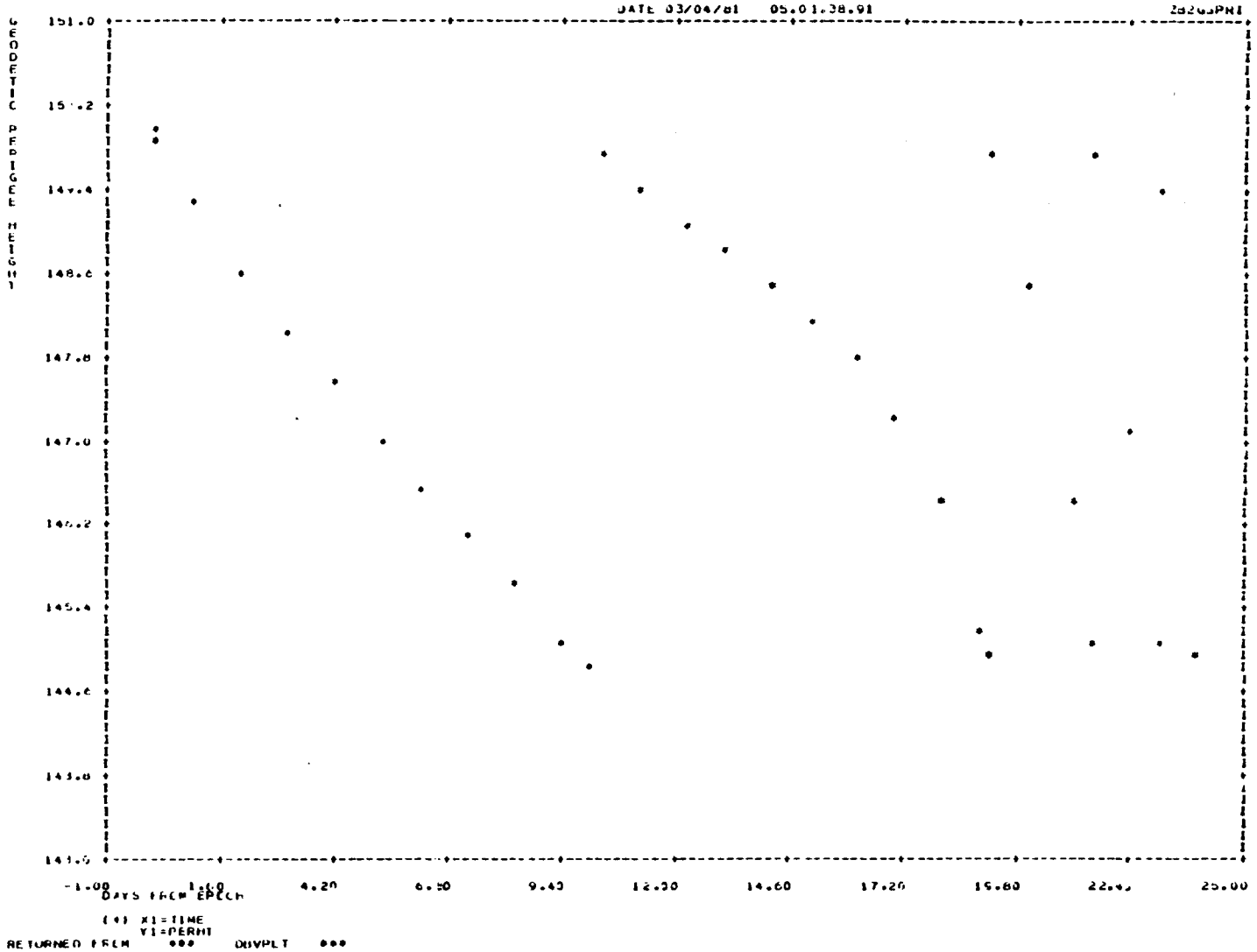


Figure 4-6. Problem 7 Plots (1 of 2)

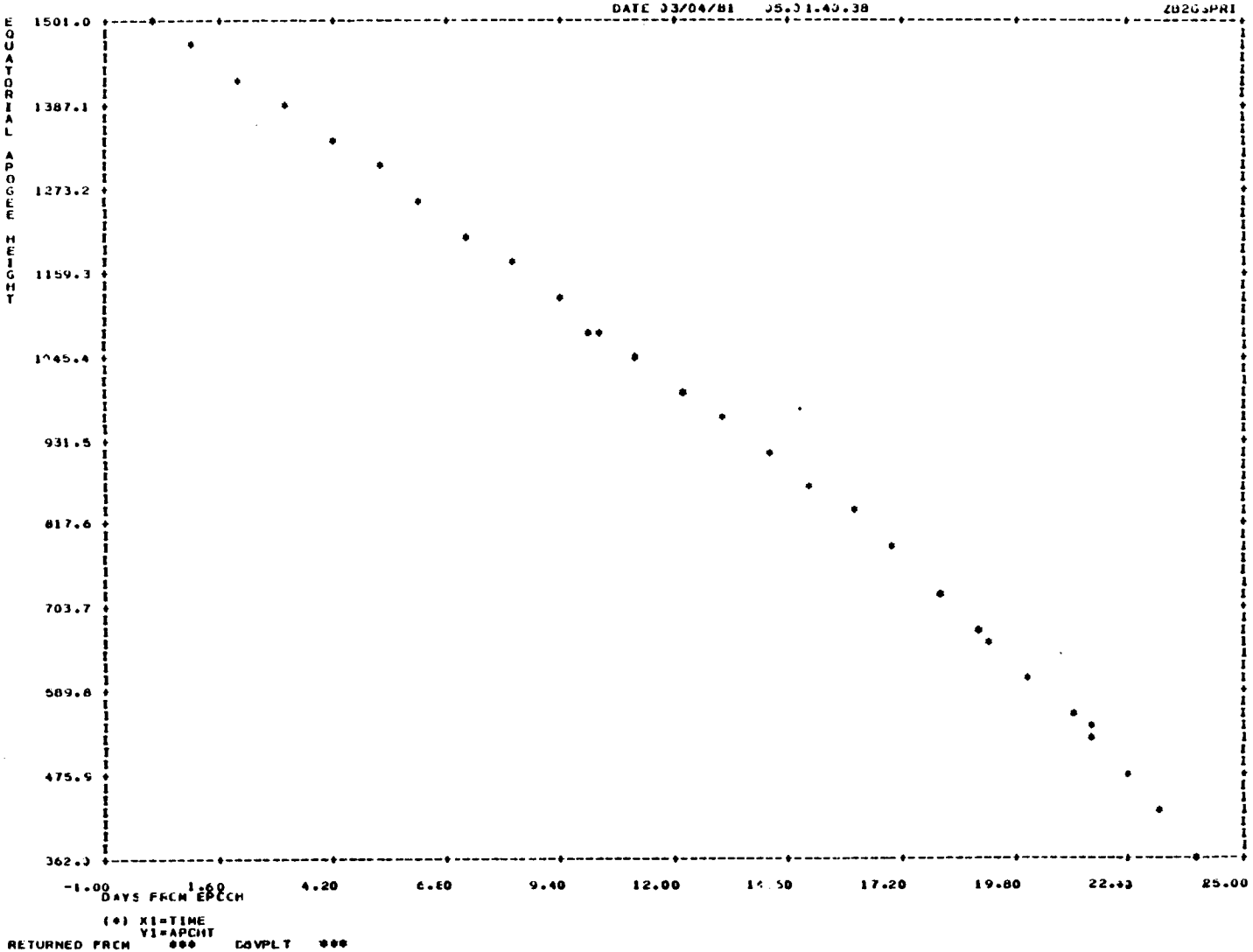


Figure 4-6. Problem 7 Plots (2 of 2)

SECTION 5 - SPECIAL USES OF GMAS

This section contains a collection of automatic sequences that perform various tasks. These automatic sequences are valuable as illustrations of various useful GMAS capabilities. This section can also be used as a basic guide to these particular GMAS capabilities, which include ORBIT File creation and reading (Section 5.1), satellite state element conversion (Section 5.2), orbital parameter comparison graph creation (Section 5.3), Monte Carlo analysis (Section 5.4), targeting and optimization (Section 5.5), and averaged orbit propagation (Section 5.6).

5.1 ORBIT FILE CREATION AND READING

5.1.1 ORBIT FILE DEFINED

An ORBIT File is a history, stored on disk or tape, of a satellite's accelerations throughout a particular propagation. In a normal orbit propagation, the orbit propagator continues until it hits a stopping condition--for example, 1 day. It then prints out various parameter information pertaining to the state of the satellite at that stopping point. The user does not, however, have access to any of the satellite states between epoch and (in this example) 1 day. An ORBIT File is a way of capturing continuously the path of the satellite during the propagation. This is accomplished by storing the satellite's acceleration components at short time intervals throughout its propagation. These accelerations are stored on a disk or tape file. Once an ORBIT File has been created, satellite elements can be retrieved at any point within the ORBIT File span without running the propagator again. ORBIT files are particularly useful in cases in which many arbitrary points in the propagation of an orbit are to be used in a program. Examples of such use include orbital parameter comparison graphing and averaged element conversion processes.

5.1.2 WRITING AN ORBIT FILE

To construct a deck that will create an ORBIT File, the user should proceed as follows:

1. Include the following JCL cards after the // EXEC GMAS,... card in the standard input deck (see Figure 2-1):

```
//FT12F001 DD DSN=&&ORBIT,UNIT=(DISK,3),DISP=(NEW,PASS),
//     DCB=(RECFM=VS,LRECL=1096,BLKSIZE=1100),
//     SPACE=(CYL,(5,5),RLSE)
```

These cards provide for a place on a disk where an ORBIT File containing up to about 4 months' worth of data can be stored. When this ORBIT File is created, it will be written out to a disk and have the name &&ORBIT. This is a temporary file and will be deleted when the GMAS run is over. If the user wishes to make the file a permanent one that can be used at a later date, he must make changes to the card formats specified above. The cards necessary to create a permanent file on disk are as follows:

```
//FT12F001 DD DSN=userid.OBS.DATA,UNIT=DISK,VOL=SER=DISKxx,
//     DISP=(NEW,CATLG),DCB=(RECFM=VS,LRECL=1096,BLKSIZE=1100),
//     SPACE=(CYL,n)
```

where userid is the user's computer identification (which must be the same as the ID on the first card of the input deck (see Figure 2-1)); xx is the identification code for the disk that is to contain the ORBIT File; and n is the amount of space (in cylinders) required on the disk for this file. Although the size of n varies, generally 20 cylinders of space is required for each month of propagation using the Cowell propagator with a 100-second step size. (For example, with a 100-second step size, for a 3-month file, n = 60; with a 200-second step size, for a 1-month file, n = 10.) The card required to create a permanent file on tape is as follows:

```
//FT12F001 DD DSN=userid.DBS.DATA,VOL=SER=tapeid
```

where userid is the user's ID, and tapeid is the tape identification number.

2. Include the Dynamics File update utility, DYNUPD, in the automatic sequence.

3. Include the following variables and their values in NAMELIST DYNUP (see words 60 through 64, record 3, of Table D-2 in the User's Guide):

Variable	Dimension	Type	Description	Default
YMDIN	1	R*8	Year, month, and day of beginning of ORBIT File	700101.1
HMSIN	1	R*8	Hour, minute, and second of beginning of ORBIT File	120000.0
YMDFN	1	R*8	Year, month, and day of end of ORBIT File	990101.0
HMSFN	1	R*8	Hour, minute, and second of end of ORBIT File	120000.0
IORBFI	1	I*4	Flag indicating whether an ORBIT File is to be written	2

Variables YMDIN, HMSIN, YMDFN, and HMSFN must be included. These are of the form YYMMDD.,HHMMSS.S, where

YY = last two digits of the year (e.g., 80 of 1980)

MM = two digits representing the month (e.g., 06 for June)

DD = two digits representing the day

HH = two digits representing the hour

MM = two digits representing the minute

SS.S = three digits representing the second and decimal second

4. Set IORBFI=1 to indicate that an ORBIT File is being created.

5. Include the ORBINP card along with the orbital information under NAMELIST ORBIN. The orbit propagation time interval must overlap the ORBIT File interval given in NAMELIST DYNUP. Set PROPM='COWELL' or 'TRCOWL'. Set IDYN=2 to include the Dynamics File update.

After performing steps 1 through 5, the card deck input should be in the following form:

```
//userid JOB...
//*FORMAT...
//*FORMAT...
// EXEC GMAS,...
//FT12F001 DD DSN=...
//
//
//GO.DATA5 DD *
 &CONTRL...
 &GMASEX...
DRIVE1
PRFCON
DYNUPD
 &DYNUP
  YMDIN=...,HMSIN=...,
  YMDFN=...,HMSFN=...,
  IORBFI=1,
  (dynamic variables)
 &END
 &DYNSEC
  (dynamic variables)
 &END
ORBINP
 &ORBIN
  IELEM=(same as YMDIN above),(same as HMSIN above),
        (elements),
    :
    :
  (input variables)
  ISTOP=1,998
  STPVAL=(same as YMDFN),(same as HMSFN + H)
    :
    :
 &END
ORBIT
EOF
```

Section 9.4 (Case 4) in the User's Guide provides an example of an automatic sequence to create an ORBIT File.

5.1.3 READING AN ORBIT FILE

To build a deck that will read an ORBIT File, the user should proceed as follows:

1. If a temporary ORBIT File was created earlier in the same run by the method described in Section 5.1.2, go to step 2. If the ORBIT File was created previously on disk and stored by the method described in Section 5.1.2, include the following card after the // EXEC GMAS,... card:

```
//FT12F001 DD DSN=userid.OBS.DATA,DISP=SHR
```

If the ORBIT File was created previously on tape and stored by the method described in Section 5.1.2, include the following card after the // EXEC GMAS,... card:

```
//FT12F001 DD DSN=userid.OBS.DATA,VOL=SER=tapeid
```

2. Include the ORBINP card. There are nonstandard uses of the variables in NAMELIST ORBIN for reading orbit files. The following variables must be included:

Variable	Value/Explanation
ICENT	=0 (to indicate that a sequential file is to be read)
ELEM	=&YYMMDD.,HHMMSS.S,6*0., where YYMMDD. is the year, month, and day of the beginning of retrieval, and HHMMSS.S is the hour, minute, and second of the beginning of retrieval. The date specified by ELEM must fall between the dates of the ORBIT File (given under DYNUPD)
PROPM	='FROG' (File Retrieval Orbit Generator)
H	=step size of 'FROG'
ISTOP	=1,998,...
STPVAL	=(same as YMDFN),(same as HMSFN),... [a]
ITERM	=1

[a] It is advisable to include the end of the file time as a stopping condition. If this is not done, 'FROG' may "jump off" the end of the file before completing stopping conditions.

After performing steps 1 through 3, the card ceck input should be in the following form:

```
//userid JOB...
//*FORMAT...
//*FORMAT...
// EXEC GMAS,...
//FT12F001 DD DSN=...
//
//
//GO.DATAS DD *
 &CONTROL...
 &GMASEX...
DRIVE1
PRFCON
ORBINP
 &ORBIN
  ICENT=0,
  ELEM=YYMMDD.,HHMMSS.S,6*0.,
  PROPM='FROG',H=0.,
  ISTOP=1,998,...
  STPVAL=(same as YMDFN),(same as HMSFN),...
  (other desired stopping variables)
  ITERM=1,
 &END
ORBIT
EOF
```

Section 9.5 (Case 5) of the User's Guide provides an example of an automatic sequence to read an ORBIT File.

5.2 SATELLITE STATE ELEMENT CONVERSION

Conversion of satellite elements from one state to another (e.g., Cartesian mean of 1950.0 to Keplerian true of date) is often desirable. To accomplish this, the user must include an ORBINP card and the following variables in NAMELIST ORBIN:

Variable	Value/Explanation	Reference
ELEM	Epoch and elements to be transformed	User's Guide, Table C-1a
IELEM	Type of input elements	

Variable	Value/Explanation	Reference
ICENT	Central body of input coordinate system	User's Guide, Table C-1a
ICORD	Input coordinate system	
IPOPT	= 99 (to prevent a call to the propagator, which is not needed in this case)	User's Guide, Table C-1b
NOUT	=n,y,z, where n denotes the output level desired, y denotes the output coordinate system, and z denotes the central body of the output coordinate system	User's Guide, Table C-1d
IPARMS	=up to 10 elements (for n=4, 5, or 6)	User's Guide, Table C-6c

An example of satellite state element conversion using GMAS is provided below.

PROBLEM: Given the following satellite state, convert this state to Earth-centered true Earth equator and equinox of date Keplerian elements with mean anomaly and print out the results:

Epoch:	January 27, 1980; 02:05:02 GMT
Coordinate system of input:	Earth-centered mean of 1950.0 Earth equator and equinox
Elements:	Cartesian: x = 6680 kilometers y = 0 kilometers z = 0 kilometers $\dot{x}$ = 0 kilometers per second $\dot{y}$ = 7.951 kilometers per second $\dot{z}$ = 2.894 kilometers per second

AUTOMATIC SEQUENCE: The automatic sequence for performing this transformation is as follows:

```
DRIVE1
PRFCON
ORBINP
 &ORBIN
  IELEM=1,ICORD=1,
  ELEM=800127.,20502.,
       6680.,0.,0.,
       0.,7.951,2.894,
```

```
  IPOPT=99,
  NOUT=4,2,1,
  IPARMS=1,2,3,4,5,20,
 &END
ORBIT
EOF
```

OUTPUT: Figure 5-1 shows the printed output resulting from the execution of the preceding automatic sequence.

5.3 ORBITAL PARAMETER COMPARISON GRAPH CREATION

Using a special GMAS parameter module, COMPLM (instead of GPARM), in conjunction with various utilities, the user can create orbital parameter comparison graphs over a given time span. This is accomplished by comparing parameters calculated from an ORBIT File to parameters generated at each step of the propagator. The parameter differences are stored in dynamic arrays, and the PLTDYN utility is used to plot their graphs. Figure 5-2 illustrates the logical flow of the orbital parameter comparison graph automatic sequence.

To set up an orbital parameter comparison run, the user should proceed as follows:

1. Determine the necessary information. The following list specifies the information that must be determined along with the values for this information used in the automatic sequence example presented in this section:

Information	Value in Example
Comparison span times:	
Beginning time	February 10, 1981; 05:30
End time	February 20, 1981; 18:00
Orbital information for two orbits (e.g., propagator, state, dynamics)	-
Time increment size (Δt) (points are plotted every Δt seconds)	3600 seconds

```
                                    GMAS PARAMETER REPORT                                    PAGE    1

DATE= 800127    2 3    2.000  ( JULIAN=2444265.58613 )   TIME FROM EPOCH=    0 DAYS  0 HRS  0 MIN  0.000SEC     TIME
CENTRAL BODY=  EARTH          COORDINATE SYSTEM= MEAN EQUATE                LOCAL TIME= 16 HRS  5 MIN   2.000 SEC
 SMA  8348.029419        ECC  .1998111573        INC  19.9982724        RAN  359.9255265        AP  .4870483478        MA  360.0000030
```

Figure 5-1. Prirted Output From Element Conversion Run

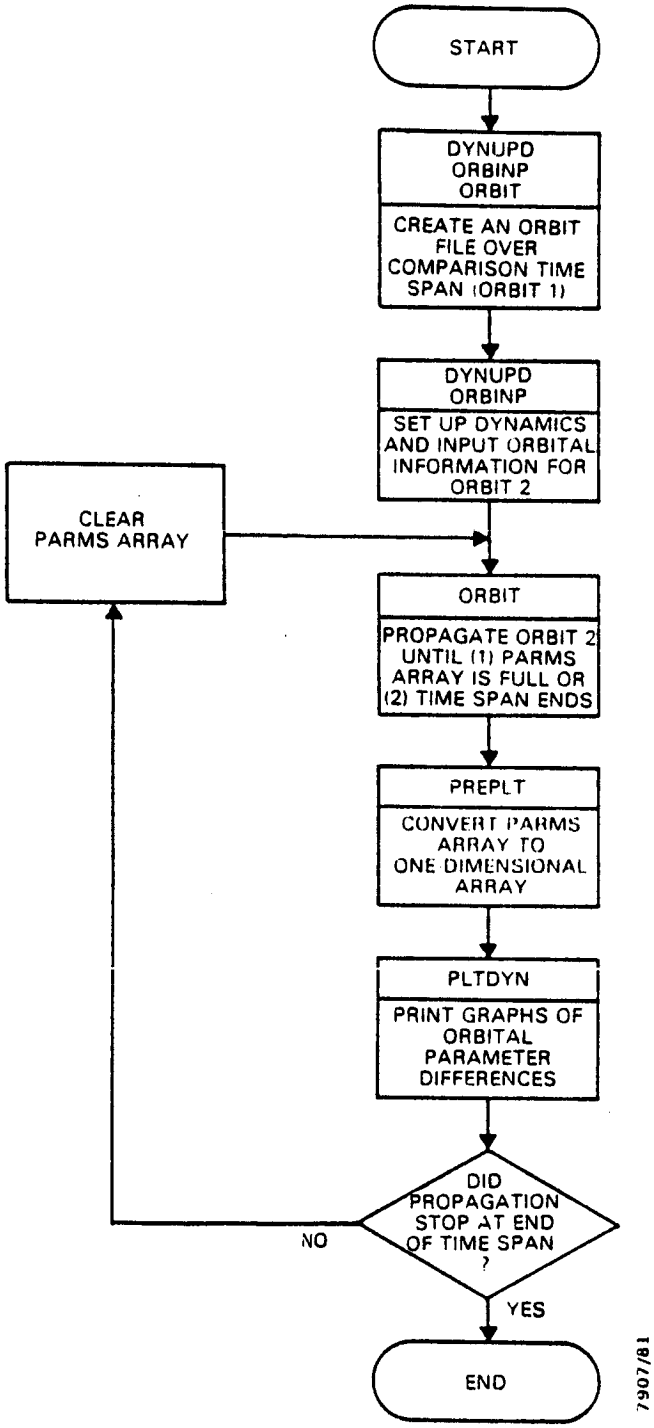


Figure 5-2. Logical Flow of Orbital Element Comparison Graph Automatic Sequence

Information	Value in Example
Comparison quantities (from Table C-4 of the User's Guide)	Semimajor axis differences (7) (kilometers); eccentricity differences (8)

2. Include the following JCL card after the // EXEC GMAS,... card in the standard input deck (see Figure 2-1):

```
//FT12F001 DD DSN=&&ORBIT,UNIT=(DISK,3),DISP=(NEW,PASS),
//     DCB=(RECFM=VS,LRECL=1096,BLKSIZE=1100),
//     SPACE=(CYL,(5,5),RLSE)
```

3. Use the automatic sequence presented following this procedure as a prototype, filling in the underlined areas with the necessary specific-case-dependent information (from step 1). This example was developed for plotting differences in two parameters. It will produce two parameter graphs (difference in semimajor axis (Δa) versus time and difference in eccentricity (Δe) versus time). Input for parameter comparison graphs is specified below; the circled letter by each item is repeated in the automatic sequence example at the corresponding point.

(a) Input the numbers (starting at the fourth element) corresponding to the desired output graphs from Table C-4 of the User's Guide.

(b) Allocate a dynamic array for each comparison parameter from step (a).

(c) Include indexes of the positions of these parameters in the PARMS array (see Section C.4.2.2 of the User's Guide).

(d) Include the names of the dynamic arrays allocated in step (b).

(e) Include a PLTDYN utility for each comparison parameter with the parameter's dynamic array name at position (f) in the example.

Two additional points must be noted here:

- If the user is using a preexisting ORBIT File for his comparison, he must (1) include an FT12F001 card indicating where the file is located (see Section 5.1.3 of this document) and (2) omit the first DYNUPD, ORBINP, and ORBIT utilities in the example automatic sequence.

- Radial, cross-track, along-track, and total errors are always computed and placed in position columns 3, 4, 5, and 6, respectively, in the PARMS array. Graphs of these can be created using the procedure described above, except the user does not need to add numbers in part a of step 3.

The automatic sequence for a comparison run is presented below. It is to be noted that spaces in the sequence column are used to clarify the association of sequence entries and explanatory comments and do not indicate a break in the sequence or the use of blank cards.

Automatic Sequence	Explanation
DRIVE1	
PRFCON	
*PARMS,A,R8,2,11,45	
*TIME,A,R8,1,45	
*DELA,A,R8,1,45	
*DELE,A,R8,1,45	
DYNUPD	
&DYNUP	
IORBFI=1,	
YMDIN=810210.,HMSIN=53000.,	Comparison begin time
YMDFN=810220.,HMSFN=180000.,	Comparison end time
(orbit 1 dynamics variables)	
&END	
&DYNSEC	
(orbit 1 dynamics variables)	
&END	
ORBINP	
&ORBIN	
IELEM=1,ICORD=2,ICENT=1,	
ELEM=810210.,53000.,	Comparison begin time (epoch)

Automatic Sequence	Explanation
(orbit 1 elements) PROPM='TRCOWL',H=100., ISTOP=1,998, STPVAL=810220.,180000., IDEL=1,IDYN=2, &END ORBIT DYNUPD &DYNUP IORBFI=2, (orbit 2 dynamics variables) &END &DYNSEC (orbit 2 dynamics variables) &END ORBINP &ORBIN IELEM=1,ICORD=2,ICENT=1, ELEM=810210.,53000.,	Comparison end time Comparison begin time (epoch)
PROPM='ANALYT',H=100.,IPOPT=2, ISTOP=1,998,1, STPVAL=810220.,180000.,3600.,	Comparison end time and time increment size
IREPT=0,0,1000, ITERM=1, PARM='COMPLM', IPARMS=12,0,0,7,8, (a)	Comparison quantity numbers from Table C-4 of User's Guide
NOUT=3, IDYN=2, &END LABEL A ORBIT *NPLOT,I,I4,1,11 &VALUE D=11*0, &END PREPLT &PREPL (b) INDEX=1,8,9, (c) XNAME='TIME','DELA','DELE', (d) &END	Initialize NPLOT

Automatic Sequence	Explanation

```
PLTDYN                                          DELA plot   (e)
 &PLTDY
  NYARRS=1,
  TITLE='SEMI-MAJOR AXIS DIFFERENCES',
  XTITLE='ELAPSED TIME (SECONDS)',
  YTITLE='SEMI-MAJOR AXIS DIFFERENCES',
  XARRAY='TIME',
  YARRAY='DELA',    (f)
  IPLOT=1,IGRID=0,
 &END
PLTDYN                                          DELE plot
 &PLTDY
  NYARRS=1,
  TITLE='ECCENTRICITY DIFFERENCES',
  XTITLE='ELAPSED TIME (SECONDS)',
  YTITLE='DELTA E',
  XARRAY='TIME',
  YARRAY='DELE',
  IPLOT=1,IGRID=0,
 &END
IF(STATEI(2).NE.1) GO TO A
EOF
```

The following points should be noted concerning this automatic sequence:

- First, the PARMS, TIME, DELA, and DELE arrays are allocated. The PARMS array is used to pass comparison values from the orbit propagator utility, ORBIT. TIME, DELA, and DELE are one-dimensional arrays that are used in the plotting utility (PLTDYN).

- The DYNUPD, ORBINP, and ORBIT utilities that follow the dynamic array allocation cards are called to create an ORBIT File over the comparison interval. The variable values for the creation of an ORBIT File are discussed in Section 5.2. IDEL under NAMELIST ORBIN is set to 1, which will cause the propagator to be deleted from core after the file has been created. This is important, especially if the user is comparing orbits using different propagators.

- The DYNUPD utility must be called to set up the dynamics for the second orbit and to set IORBFI=2 since the user does not wish to create an ORBIT File on the next call to the ORBIT utility.

- ORBINP is called to input the elements and propagator associated with the second orbit. The stopping conditions are set to stop periodically (in this case, every 3600 seconds) until the end of the comparison span. Selected parameter differences will be calculated and stored at each stop to be graphed later in the automatic sequence. Section C.2.5 of the User's Guide describes the other non-standard input through NAMELIST ORBIN.

- LABEL A is used in the looping procedure.

- The ORBIT utility is called to propagate the orbit, stopping every 3600 seconds. ORBIT will continue until either the PARMS array is full or the orbit propagation has reached the end of the comparison interval.

- Dynamic array NPLOT must be initialized before the call is made to the PREPLT utility.

- PREPLT is called to transfer the information from the PARMS array into the one-dimensional arrays TIME, DELA, and DELE, which were allocated at the beginning of the automatic sequence. Section 2.4 of the Software Resources document describes PREPLT.

- PLTDYN is called to generate the desired comparison plots. This utility must be called once for each parameter being compared. Section 5.3.9 of the User's Guide describes PLTDYN.

- If the orbit propagator stops only because the PARMS array is full, it will set STATEI(2)=2. If the orbit

propagator hits a terminal stopping condition (end of the comparison span), it will set STATEI(2)=1. If the latter is the case, the run will end. Otherwise, processing will continue at LABEL A. The PARMS array will be reinitialized, and the orbit propagator utility, ORBIT, will be called again to continue propagating.

Figures 5-3 and 5-4 show some of the printed output resulting from a run made using the automatic sequence format described above. This run compares the orbital parameters generated using a time-regularized Cowell propagator with drag as a dynamic effect with the parameters generated using an analytic propagator without drag as a dynamic effect. In this particular example, the PARMS array fills up six times, thus creating six comparison graphs of each of the two parameters compared. Figures 5-3 shows the first GMAS comparison results table; Figure 5-4 shows the corresponding semimajor comparison graph. Section 9.6 (Case 6) of the User's Guide and Section 2.4.4 of the Software Resources document provide other examples of parameter comparison.

5.4 MONTE CARLO ANALYSIS

The Monte Carlo Program driver, MONDRV, provides the user with a generalized Monte Carlo error analysis capability that can be applied to any user-defined funtion. In Monte Carlo analysis, the function is evaluated (sampled) a large number of times, with each sample being supplied a set of control (independent) parameters whose values are the nominal values plus perturbed values obtained from a set of random numbers. The output (dependent) parameters of the function provide the raw data for the statistical and probabilistic analysis. Figure 5-5 shows the general flow of the Monte Carlo Program along with the input variables connected with each stage of the processing.

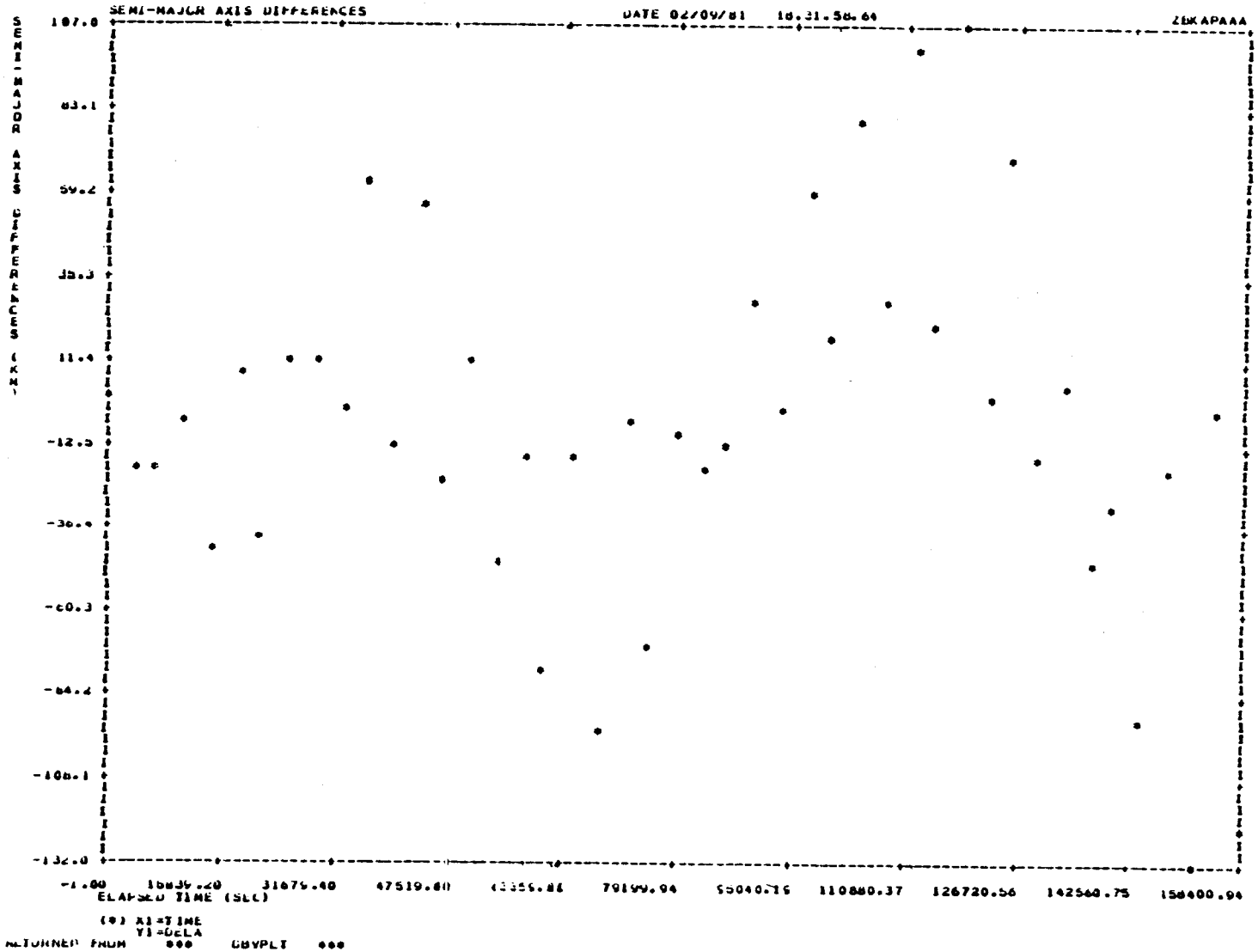


Figure 5-3. GMAS Comparison Results

YYMMDD	HHMMSS	EVENT	D	TIME FROM EPOCH HRS	TRUE ANOMALY (DEG)	*RADIAL* POS. ERROR (KM)	CROSS-TRACK POS. ERR (KM)	IN-TRACK POS. ERR (KM)	TOTAL POS. ERROR (KM)	TIMESEP (SEC)	DEL A (METER)	DEL G (DEG)
810210.	83000.0	TIME	0	0.0	360.00	-0.000	-0.000	0.000	0.000	0.0	-0.000	-0.000
810210.	93000.0	TIME	0	1.0	185.54	-0.093	0.024	0.387	0.399	0.1	-20.319	0.002
810210.	103000.0	TIME	0	2.0	17.17	-0.073	-0.017	0.750	0.754	0.1	-19.575	0.007
810210.	113000.0	TIME	0	3.0	196.68	0.012	0.071	1.164	1.166	0.2	-7.009	0.005
810210.	123000.0	TIME	0	4.0	33.96	-0.130	-0.087	1.352	1.361	0.2	-43.402	0.009
810210.	133000.0	TIME	0	5.0	208.00	0.121	0.163	1.427	1.442	0.2	9.228	0.006
810210.	143000.0	TIME	0	6.0	50.13	-0.153	-0.117	1.616	1.628	0.2	-39.820	0.008
810210.	153000.0	TIME	0	7.0	219.62	0.147	0.126	1.319	1.333	0.2	10.469	0.004
810210.	163000.0	TIME	0	8.0	65.59	-0.150	-0.024	1.760	1.766	0.2	10.674	0.008
810210.	173000.0	TIME	0	9.0	231.63	0.063	-0.079	1.468	1.473	0.2	-2.272	0.001
810210.	183000.0	TIME	0	10.0	80.34	-0.131	0.166	2.149	2.158	0.3	61.661	0.009
810210.	193000.0	TIME	0	11.0	244.13	0.010	-0.307	2.119	2.142	0.3	-14.067	-0.002
810210.	203000.0	TIME	0	12.0	94.42	-0.055	0.304	2.760	2.778	0.4	54.926	0.006
810210.	213000.0	TIME	0	13.0	257.19	-0.025	-0.410	2.832	2.862	0.4	-24.223	-0.004
810210.	223000.0	TIME	0	14.0	107.86	-0.063	0.326	3.296	3.313	0.5	10.085	0.002
810210.	233000.0	TIME	0	15.0	270.85	-0.029	-0.369	3.310	3.330	0.5	-45.868	-0.001
810210.	243000.0	TIME	0	16.0	120.71	-0.061	0.243	3.698	3.707	0.5	-17.619	0.003
810210.	253000.0	TIME	0	17.0	285.17	0.013	-0.246	3.764	3.772	0.5	-77.851	0.005
810210.	263000.0	TIME	0	18.0	133.02	-0.150	0.138	4.062	4.067	0.6	-14.323	0.008
810211.	3000.0	TIME	0	19.0	300.18	0.094	-0.103	4.313	4.315	0.6	-94.313	0.010
810211.	13000.0	TIME	0	20.0	144.66	-0.213	0.026	4.411	4.417	0.7	-4.682	0.013
810211.	23000.0	TIME	0	21.0	315.92	0.143	0.040	4.842	4.844	0.6	-70.875	0.012
810211.	33000.0	TIME	0	22.0	156.34	-0.229	-0.137	4.886	4.893	0.8	-7.513	0.015
810211.	43000.0	TIME	0	23.0	332.37	0.102	0.175	5.395	5.399	0.7	-19.928	0.015
810211.	53000.0	TIME	1	0.0	167.57	-0.175	-0.295	5.656	5.667	0.9	-13.050	0.018
810211.	63000.0	TIME	1	1.0	349.36	-0.004	0.262	6.071	6.076	0.8	27.515	0.020
810211.	73000.0	TIME	1	2.0	176.67	-0.038	-0.335	6.505	6.514	1.0	-3.552	0.022
810211.	83000.0	TIME	1	3.0	6.58	-0.122	0.266	6.699	6.705	0.8	59.134	0.027
810211.	93000.0	TIME	1	4.0	189.76	0.149	-0.280	6.943	6.950	1.1	17.668	0.023
810211.	103000.0	TIME	1	5.0	23.63	-0.230	0.249	7.025	7.033	0.9	80.102	0.029
810211.	113000.0	TIME	1	6.0	200.96	0.255	-0.266	6.867	6.879	1.1	29.539	0.021
810211.	123000.0	TIME	1	7.0	40.18	-0.318	0.328	7.132	7.147	0.9	99.881	0.028
810211.	133000.0	TIME	1	8.0	212.39	0.325	-0.496	6.810	6.836	1.0	20.886	0.018
810211.	143000.0	TIME	1	9.0	56.07	-0.375	0.515	7.410	7.437	1.0	106.597	0.027
810211.	153000.0	TIME	1	10.0	224.16	0.294	-0.753	7.242	7.287	1.1	2.609	0.016
810211.	163000.0	TIME	1	11.0	71.26	-0.392	0.679	7.589	8.028	1.1	68.584	0.023
810211.	173000.0	TIME	1	12.0	236.35	0.275	-0.882	7.909	7.963	1.2	-16.006	0.014
810211.	183000.0	TIME	1	13.0	85.77	-0.365	0.696	8.581	8.618	1.2	3.291	0.018
810211.	193000.0	TIME	1	14.0	249.05	0.262	-0.830	8.385	8.430	1.2	-45.628	0.015
810211.	203000.0	TIME	1	15.0	99.62	-0.395	0.564	8.993	9.020	1.3	-30.527	0.018
810211.	213000.0	TIME	1	16.0	262.33	0.306	-0.668	8.775	8.806	1.2	-92.718	0.022
810211.	223000.0	TIME	1	17.0	112.85	-0.444	0.375	9.312	9.330	1.4	-20.373	0.023
810211.	233000.0	TIME	1	18.0	276.22	0.365	-0.465	9.299	9.317	1.3	-131.751	0.030
810212.	3000.0	TIME	1	19.0	125.49	-0.493	0.180	9.602	9.617	1.4	-2.025	0.029
810212.	13000.0	TIME	1	20.0	290.78	0.406	-0.249	9.844	9.856	1.3	-120.055	0.033

Figure 5-4. GMAS Comparison Graph

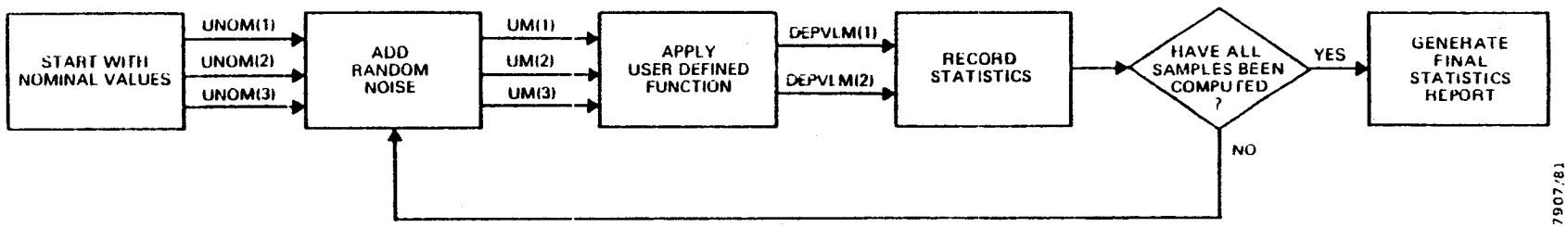


Figure 5-5. Monte Carlo Program General Flow and Basic Input Variables

Section 5.1.2 of the User's Guide provides a complete description of the GMAS Monte Carlo usage and input. To use the Monte Carlo analysis capability, the user should proceed as follows:

1. Create a Monte Carlo automatic sequence. The form of the automatic sequence is as follows:

```
MONDRV
 &MONDR
  (Monte Carlo variables)
 &END
PRFCON (or TARCON)        }   User-defined function
  (utilities)             }
EOF
```

MONDRV must be used instead of DRIVE1 and must be followed by the Monte Carlo input variables under NAMELIST MONDR. The user-defined function, which is included next, must begin with a controller (either PRFCON or TARCON). The user-defined "function" is actually a partial automatic sequence that uses values from dynamic array UM (independent variables) to produce dependent values that are stored in dynamic array DEPVLM.

2. Determine the values of the variables under NAMELIST MONDR and include them in the automatic sequence. Table 5-2 presents a categorical list of the Monte Carlo variables and a brief description of each. In this list, the variables underlined are those usually required in a Monte Carlo run. Appendix A of the User's Guide provides a complete list of Monte Carlo variables that includes dimension, type, and default values.

3. Develop the user-defined function. First, define the function to be used by placing the appropriate controller, executive utility, and utility load module calls in the automatic sequence following NAMELIST MONDR. (User-supplied utilities may also be used. Section 5 of this

Table 5-2. Monte Carlo Program Variables

VARIABLE	DESCRIPTION
	NOMINAL VALUES
NINDV	NUMBER OF INDEPENDENT VARIABLES
UNOM	NOMINAL VALUES OF INDEPENDENT VARIABLES
	RANDOM PERTURBATION PARAMETERS
COVIN	COVARIANCE MATRIX FOR INDEPENDENT VARIABLES (SET ICORIN=1)
ICORIN	FLAG TO INDICATE WHETHER INDEPENDENT VARIABLES ARE CORRELATED: = 0, UNCORRELATED (DEFAULT) (SIGMA USED) = 1, CORRELATED (COVIN USED)
IDIST	DISTRIBUTION FOR ITH IDEPENDENT VARIABLE
SEED	SEED OF THE RANDOM NUMBER GENERATOR
SIGMA	σ VALUES FOR INDEPENDENT VARIABLES IF THEY ARE UNCORRELATED
	DEPENDENT VARIABLES
DEPTL	LOWER AND UPPER TOLERANCES FOR EACH DEPENDENT VARIABLE
IDEPVL	FLAG TO SELECT DEPENDENT VARIABLE CALCULATION METHOD
NDEPV	NUMBER OF DEPENDENT VARIABLES
	DEPENDENT VARIABLE STATISTICS OUTPUT
HSTGRM	NUMBER OF HISTOGRAM BREAK POINTS AND HISTOGRAM VALUES FOR EACH DEPENDENT VARIABLE (SET IHIST≠0)
IHIST	FLAG TO REQUEST HISTOGRAM OUTPUT
ICOV	COVARIANCE MATRIX OUTPUT
IDEB	DEBUG OUTPUT
ISMPLP	PRINT MONTE CARLO RANDOM NUMBERS
ITOL	PRINT TOLERANCE INFORMATION
NPRNT	PRINT FREQUENCY
TITLE	DATA BASE TITLE TO BE PRINTED AT TOP OF EACH PAGE
	GENERAL VARIABLES
NMCS	NUMBER OF MONTE CARLO SAMPLES
INITM	INITIALIZATION FLAG FOR INDEPENDENT VARIABLES

7907/81

document provides the information necessary for creating utilities.) Then provide for interfacing the dynamic arrays UM and DEPVLM (see Table A-2 of the User's Guide) with the user-defined function. This can be accomplished by (a) making the input and output variables in the user-defined function dynamic arrays, thus enabling the use of utility DSTORE (see Section 5.3.2 of the User's Guide) to transfer the data between UM and DEPVLM and the user-defined dynamic arrays, or (b) using the executive service routines FECHDA and STORDA (see Appendix E of the User's Guide) from within the function to access and store UM and DEPVLM directly. In general, method a provides more flexibility than method b. With method a, the problem can be changed via user input; method b requires a coding change to alter the problem setup.

Section 9.10 (Case 10) of the User's Guide provides a problem example of the use of the GMAS Monte Carlo error analysis capability.

5.5 TARGETING AND OPTIMIZATION

The targeting and optimization controller, TARCON, provides the user with a numerical targeting and optimization capability that can be applied to any user-defined function composed of a series of utilities. The GMAS targeting and optimization capability can be used in the selection of certain mission control parameters (e.g., ΔX, ΔY, ΔZ) to satisfy mission constraints (e.g., raise perigee height while maintaining the argument of perigee) while optimizing some mission objective (e.g., minimize the magnitude of the ΔV vector). TARCON varies the control parameters until the constraints are satisfied to within their tolerances and the optimization variable is minimized (or maximized).

Figure 5-6 specifies the primary variables used in the targeting and optimization process. The object in targeting and optimization is to search through all possible values of the control parameters (U) to find the set of parameters that, when processed by the user-defined function to produce the target variable DEPVL, will give the desired target values (DEPVAL) within their tolerances (DEPTL) while optimizing the optimization variable (OPTVL).

To use the GMAS targeting and optimization capability, the user should proceed as follows:

1. Create a targeting and optimization automatic sequence. The form of the automatic sequence is as follows:

```
DRIVE1
TARCON
 &TARGC
  (targeting and optimization variables)
 &END
 (user-defined function)
EOF
```

It is to be noted that DRIVE1 is followed by the targeting and optimization controller (TARCON) instead of PRFCON.

2. Determine the values of the variables under NAMELIST TARGC and include them in the automatic sequence. Table 5-3 presents a categorical list of some of the targeting and optimization variables that are frequently used. In this list, the variables underlined are those which must be included in any targeting run. Appendix B of the User's Guide provides a complete list of targeting and optimization variables.

3. Develop the user-defined function to be targeted. First, define the function to be used by placing the appropriate utility load module and executive utility calls in the automatic sequence following NAMELIST TARGC. The end of the user-defined function is indicated by a driver or

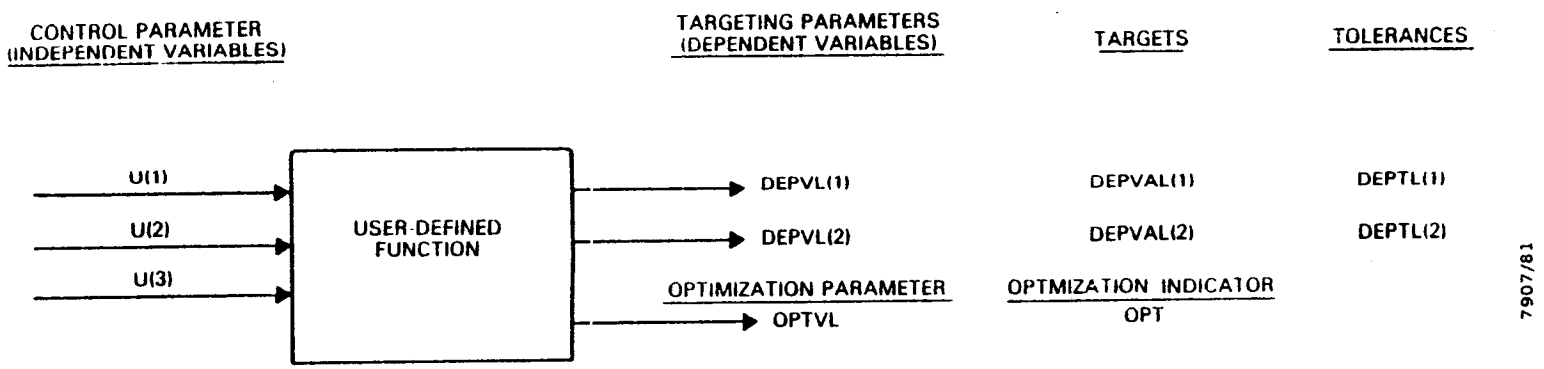


Figure 5-6. Primary Targeting and Optimization Variables

Table 5-3. Frequently Used Targeting and Optimization Variables (1 of 2)

NAME	DIMENSION	TYPE	DESCRIPTION
			CONTROL (INDEPENDENT) VARIABLES
NINDV	1	I*4	NUMBER OF INDEPENDENT (CONTROL) VARIABLES
U	25	R*8	INITIAL GUESS FOR THE VALUES OF THE INDEPENDENT VARIABLES
MODLW	1	I*4	FLAG CONTROLLING THE WEIGHTING OF THE INDEPENDENT VARIABLES: = 0, USE INPUT WEIGHTING WVU (SEE BELOW) = 1, AUTOMATIC CONTROL WEIGHTING (WVU(I) = 1.0/U(I))
WVU	25	R*8	INDEPENDENT VARIABLE WEIGHTING. WVU(I) IS THE WEIGHT ASSOCIATED WITH THE ITH VARIABLE. IF TARGETING IS DONE USING THE ΔV VECTOR AS THE INDEPENDENT VARIABLE, SET WVU=1,1,1 TO EQUALLY WEIGHT THE CONTROLS
PERT	25	R*8	PERTURBATIONS ADDED TO THE INDEPENDENT VARIABLES USED TO NUMERICALLY CALCULATE THE PARTIAL DERIVATIVES. UNLESS THE FUNCTION IS UNUSUALLY FLUCTUATING, THE DEFAULT OF .0001 WILL SUFFICE
			TARGETING (DEPENDENT) VARIABLES
NDEPV	1	I*4	NUMBER OF DEPENDENT (TARGET) VARIABLES TO BE SATISFIED
DEPVAL	25	R*8	DESIRED VALUE OF THE ITH DEPENDENT (TARGET) VARIABLE
DEPTL	25	R*8	ABSOLUTE TOLERANCE TO WHICH THE ITH DEPENDENT (TARGET) VARIABLE IS TO BE SATISFIED. IF NINDV=n AND NDEPV=m, THE USER FUNCTION f: $R^n \rightarrow R^m$ IS TARGETED WHEN $\lvert f(u) - DEPVAL \rvert \leq DEPTL$
IDEPVR	25	I*4	FLAG TO INDICATE THE TYPE OF CONSTRAINT DESIRED FOR THE ITH DEPENDENT (TARGET) VARIABLE: = -1, INEQUALITY WITH LOWER BOUND = 0, EQUALITY CONSTRAINT (DEFAULT) = +1, INEQUALITY WITH AN UPPER BOUND
IFDEG	25	I*4	FLAG THAT CONTROLS THE 360-DEGREE-TO-0-DEGREE DISCONTINUITY IN THE ITH DEPENDENT (TARGET) VARIABLE: = 0, NO CORRECTION MADE (DEFAULT) = 1, IF $\lvert E(I) \rvert > 180$ DEGREES, SET E(I) = E(I) ± 180 DEGREES THIS FLAG SHOULD ALWAYS BE SET TO 1 IF THE TARGETING (DEPENDENT) VARIABLES ARE ANGULAR QUANTITIES (IN DEGREES)

7907/81

Table 5-3. Frequently Used Targeting and Optimization Variables (2 of 2)

NAME	DIMENSION	TYPE	DESCRIPTION
			OPTIMIZATION VARIABLES
OPT	1	I*4	OPTIMIZATION FLAG: = −1, MINIMIZE OPTVL = 0, TARGETING ONLY (DEFAULT) = +1, MAXIMIZE OPTVL
WOPT	1	R*8	WEIGHTING CONSTANT FOR THE OPTIMIZATION VARIABLE; SHOULD BE THE RECIPROCAL OF THE NOMINAL VALUE OF OPTVL (DEFAULT = 1.0)
			TARGETING AND OPTIMIZATION GENERAL CONTROL VARIABLES
SHCHM	1	I*4	FLAG TO SPECIFY THE ALGORITHM USED FOR THE TARGETING/OPTIMIZATION: = 0, NO TARGETING/OPTIMIZATION = 1, STEEPEST DESCENT METHOD = 2, CONJUGATE GRADIENT METHOD = 3, DAVIDON METHOD = 4, PROJECTED GRADIENT METHOD (DEFAULT) } MOST COMMONLY USED = 5, ACCELERATED PROJECTED GRADIENT METHOD }
MAXITR	1	I*4	MAXIMUM NUMBER OF ITERATIONS ALLOWED (DEFAULT = 10)

7907/81

controller call or by an end-of-file indicator. Then provide for interfacing the dynamic arrays U, DEPVL, and OPTVL with the user-defined function. This can be accomplished by (a) making the input and output variables in the user-defined function dynamic arrays, thus enabling the use of utility DSTORE (see Section 5.3.2 of the User's Guide) to transfer the data between U, DEPVL, and OPTVL and the user-defined dynamic arrays, or (b) using the executive service routines FECHDA and STORDA (see Appendix E) from within the function to access and store U, DEPVL, and OPTVL directly. In general, method a provides more flexibility than method b. With method a, the problem can be changed via user input; method b requires a coding change to alter the problem setup.

This subsection is intended to introduce the user to the basic operation of the GMAS targeting and optimization capabilities. For a more comprehensive explanation of TARCON and examples, the reader is referred to the following sources:

Subject	Reference
TARCON Controller (description)	User's Guide, Section 5.2.2
Targeting and optimization mathematics (description)	User's Guide, Appendix B
NAMELIST TARGC input (description)	User's Guide, Appendix B
Optimization of a second-degree polynomial with linear constraints (example)	User's Guide, Section 9.11 (Case 11)
Impulsive maneuver targeting, fixed time (example)	Software Resources document, Section 3.1
Impulsive maneuver targeting, variable time (example)	Software Resources document, Section 3.2

5.6 AVERAGED ORBIT PROPAGATION

The averaged orbit propagator, AVGVOP, is a rapid orbit generator, designed for moderately accurate and efficient computation of the long-term motion of artificial satellites. By averaging out short-term periodic effects, AVGVOP is able to take much larger steps than its numerical counterparts (e.g., Cowell takes 100 steps per revolution; AVGVOP takes 1 step per revolution). It does not, however, suffer from the imprecision associated with purely analytic propagators. Because of its great efficiency, AVGVOP provides a valuable mission analysis tool for studies involving long-term effects of satellite parameters.

To run the averaged orbit propagator, the user can apply the procedure for propagating orbits discussed in Section 3 of this document, noting the following:

- PROPM must be set to 'AVGVOP'.
- Input elements must be averaged elements. Averaged elements can be obtained by using the averaged element conversion utility, AVECON.
- Averaged elements must be created using the same force model as the propagator will be using.
- The step size, H, should be approximately one satellite period (in seconds).

Since the creation of averaged (mean) elements requires a good deal of utility manipulation in the automatic sequence, an automatic sequence, MEANEL, has been created expressly for the purpose of calculating mean elements. By making suitable changes to the GMAS input deck (see Figure 2-1), this automatic sequence can be brought into GMAS from a library of preexisting automatic sequences and updated to satisfy the user's specific requirements. The GMAS

automatic sequence library and updating procedures are mentioned in Section 7 of this document. Section 3.3 of User's Guide provides a complete discussion of GMAS automatic sequence card updates.

An automatic sequence, AVEGEN, has been created to convert the user's osculating elements to mean elements and then propagate these mean elements to a desired stopping condition. A thorough description of the use of AVEGEN can be found in Section 3.6 of the GMAS Software Resources document. One drawback to using AVEGEN is that the user cannot specify the epoch time of the generated mean elements. If the user wishes to fix the epoch time for the mean elements, he should use the following method:

1. Follow the steps described in Section 3.5.3 of the Software Resources document to create mean elements at the desired epoch.

2. Add the following cards to the bottom of the automatic sequence updates from step 1 in order to propagate the mean elements that have been created:

```
AVECON,1,ADD
ORBINP
 &ORBIN
  PROPM='AVGVOP',IELEM=0,
  H=(step size),
  IDYN=2,
  (other orbital input stopping conditions)
 &END
ORBIT
EOFADD
```

Section 3.5.5 of the Software Resources document provides a sample case using MEANEL.

The total deck setup for the creation of mean elements using MEANEL and the subsequent propagating of these elements is given below along with comments.

Input Cards	Explanation
`//ZBNAMAVG JOB...`	
`//*FORMAT PR,...`	
`//*FORMAT PU,...`	
`// EXEC GMAS,REGION.GO=375K`	
`//FT12F001 DD DSN=&&ORBFIL,UNIT=(DISK,3),`	Allocate orbit file space (see Section 5.2)
`//  DISP=(NEW,PASS),DCB=(RECFM=VS,LRECL=1096,`	
`//  BLKSIZE=1100,BUFNO=2),SPACE=(CYL,(5,5),RLSE)`	
`//GO.DATA5 DD *`	
`&CONTRL IFTUBE=50,IFTABL=49,IFTPRT=9,&END`	
`&GMASEX SEQNAM='MEANEL',IUPDAT=1,IBATCH=1,&END`	Use pre-existing MEANEL sequence and update
`UPDATES`	
`DYNUPD,1,UPD`	Input force model
`&DYNUP`	
`(variable)`	
`&END`	
`&DYNSEC`	
`(variables)`	
`&END`	
`EOFUPD`	
`ORBINP,1,UPD`	Input initial state and stopping condition for first propagation
`&ORBIN`	
`(variables)`	
`&END`	
`EOFUPD`	
`DYNUPD,2,UPD`	Input force model again and set start and end times of orbit file
`&DYNUP`	
`(variables)`	
`&END`	
`&DYNSEC`	
`&END`	
`EOFUPD`	

Input Cards	Explanation
ORBINP,2,UPD &ORBIN (variables) &END EOFUPD	Set stop time for second propagation
ORBINP,3,UPD &ORBIN (variables) &END EOFUPD	Set epoch for mean elements
AVECON,1,UPD &AVECO (variables) &END EOFUPD	Set number of periods to be averaged over
AVECON,1,ADD ORBINP &ORBIN (variables) &END ORBIT EOFADD	Propagate mean elements using AVGVOP. Input stopping conditions. Set IDYN=2 for dynamics

SECTION 6 - CREATION OF GMAS MODULES

Thus far in this primer, only utilities from the operational GMAS utility library have been used in automatic sequences. The real power of GMAS is that users can develop their own utilities. These utilities can be stored in libraries and used in user-defined automatic sequences, thereby enabling the user to string utilities together in any desired configuration.

This section demonstrates how the user can integrate his own software with the existing GMAS software through the use of the automatic sequence to solve mission-specific problems. Topics discussed include the creation and use of utilities (Section 6.1), the modification of existing utilities (Section 6.2), the creation of special output parameter modules (Section 6.3), the creation of parameter modules in general (Section 6.4), and the use of GMAS service routines in user load modules (Section 6.5). Since utilities are created from FORTRAN routines, the reader is assumed to be familiar with the IBM FORTRAN IV programming language.

6.1 CREATION OF UTILITIES

This section discusses the various aspects of converting a program or routine into a utility and using it in an automatic sequence. Utility input is usually through NAMELISTs or dynamic arrays. Output is usually printed out (using the FORTRAN WRITE statement) or placed in dynamic arrays. This section uses a series of examples to describe the process of creating and using a utility. The first example (Section 6.1.1) uses the most simple forms of input and output; subsequent examples (Sections 6.1.2 through 6.1.4) use more complicated input/output methods. The actual functions performed by the routines in the examples are kept simple for demonstration purposes. The first three examples involve

averaging four numbers. The fourth example involves taking the sum of three numbers.

Before studying the examples, the reader should understand the general process involved in converting a subroutine or collection of subroutines into a utility to be used in an automatic sequence. Figure 6-1 helps to illustrate this process. A special program developed for GMAS, UTLBLD, uses the user's FORTRAN subroutines (source) to build a utility (AVGUTL). It then stores the utility in a user library (ZBXXX.MYFILE.LOAD) on a disk (DISKXX).

The user procedure is as follows:

1. Determine the input/output requirements of the FORTRAN program that is to be converted to a utility. Sections 6.1.1 through 6.1.4 each present a different method of input/output. The user's program can utilize any one or any combination of these methods.

2. Write a FORTRAN program for use in creating the desired utility. This FORTRAN program must be written in a certain manner to make it compatible with GMAS software.

3. Collect the information necessary to create and store the utility on disk. The types of information required are noted below, along with the specific example of each used in Figure 6-1:

Information Item	Example in Figure 6-1
Deck of cards consisting of FORTRAN code for utility	FORTRAN source deck
Name of main subroutine in source deck	AVRAGE
Name of new utility	AVGUTL
Name of library in which utility is to be stored	ZBXXX.MYFILE.LOAD
Name of disk on which library resides	DISKXX

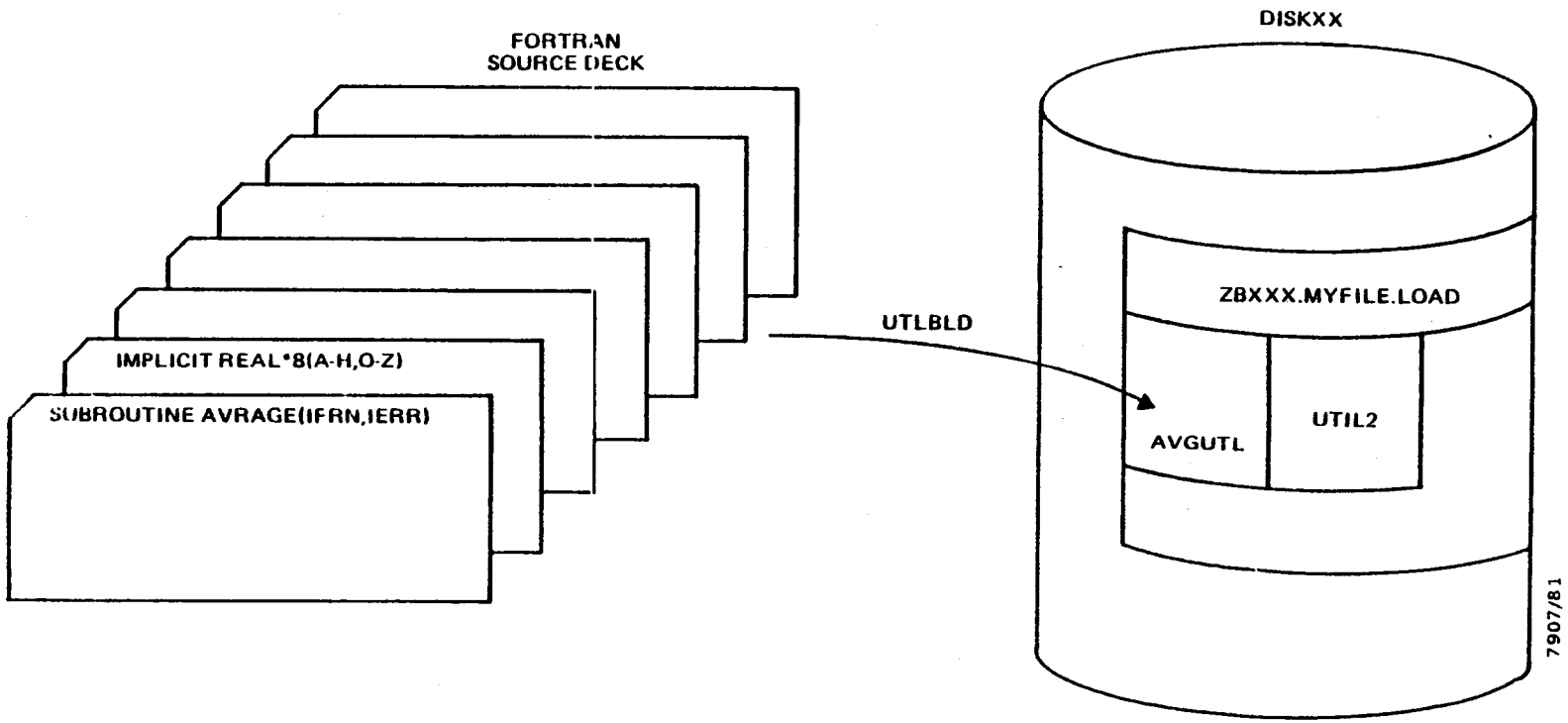


Figure 6-1. Building a Utility

4. Create a deck that will both build a utility containing this program and store the utility in the user's library.

5. Submit the deck created in step 4.

Steps 1 and 5 require no further explanation. Steps 2, 3, and 4 are discussed in detail in the following sections.

6.1.1 EXAMPLE 1: UTILITY WITH NAMELIST INPUT

6.1.1.1 Creating the Utility

The objective in this example is to create a utility named AVGUTL that reads in four R*8 numbers through NAMELIST input, calculates their average, and prints out both the four numbers and their average. The steps necessary to accomplish this are described below.

6.1.1.1.1 Writing the FORTRAN Program

The following FORTRAN program can be used to meet the objective in this example:

```
      col. 7
      ↓
      SUBROUTINE AVRAGE(IFRN,IERR)     (1)
      IMPLICIT REAL*8(A-H,O-Z)         (2)
      NAMELIST/AVGUT/A                 (3)
      DIMENSION A(4)                   (4)
      READ(IFRN,AVGUT)                 (5)
      SUM=0.0
      DO 100 I=1,4
           SUM=SUM+A(I)
  100 CONTINUE
      AVG=SUM/4.0
      WRITE(6,900)(A(I),I=1,4),AVG
  900 FORMAT('1THE AVERAGE OF ',4G10.2,' IS ',G10.2)
      RETURN                           (6)
      END
```

The following rules must be adhered to when creating a new program or converting an existing program to a GMAS utility:

- The main routine must be written as a subroutine with an argument list containing IFRN and IERR
- IFRN is the FORTRAN reference number of the executive file. This is the file that contains the user's NAMELIST input. IERR is used as an error flag. The user can set this flag to a nonzero value in his application program (utility) if an error occurs. When control is returned to the GMAS executive, the normal GMAS error procedure will be followed. (See (1) in the FORTRAN program presented above).
- Since all GMAS routines use double-precision variables, the user should include the IMPLICIT statement ((2)) in order to maintain system integrity.
- If input to the main routine is by NAMELIST under the utility card in the automatic sequence, the main routine should include the following cards:
  - NAMELIST statement ((3))--The general format of the NAMELIST statement is

    $$NAMELIST/X/a_1,a_2,a_3,a_4,\ldots,a_n$$

    where X is the NAMELIST name, and $a_1,a_2,a_3,\ldots,a_n$ are the variables or arrays in the NAMELIST.
  - DIMENSION statement ((4)) for any arrays in the NAMELIST--The NAMELIST and DIMENSION statements must occur before the first line of executable code.
  - READ statement ((5)) to read the NAMELIST input from the automatic sequence file (FORTRAN reference number = IFRN)

- The main routine code must be followed by a RETURN statement (⑥).
- No routine or subroutine in a GMAS utility should include a STOP statement. The execution of a STOP statement will terminate GMAS execution.

6.1.1.1.2 Collecting the Necessary Information

The information necessary to create and store the desired utility is as follows:

Information Item	Specific
Deck of cards consisting of FORTRAN code for utility	FORTRAN source deck
Name of main subroutine in source deck	AVRAGE
Name of new utility	AVGUTL
Name of library in which utility is to be stored	ZBXXX.MYFILE.LOAD
Name of disk on which library resides	DISKXX

6.1.1.1.3 Creating the Input Deck

The input card deck presented below can be used to build the desired utility. Underlined items indicate information that must be supplied by the user.

```
col. 1
*
//ZBXXXAVG JOB...
//*FORMAT...
//*FORMAT...
// EXEC UTLBLD
//PREP.DATA5 DD *
 &UTIL
  UTLNAM='AVGUTL',     (8)
  PRGNAM='AVRAGE',     (9)
 &END
FPARM XREF     (10)
//SOURCE.FORTIN DD *
(program cards (source code))     (11)
/*
```

```
//LINK.SYSLMOD DD DSN=ZBXXX.MYFILE.LOAD,UNIT=DISK,        (12)
//      VOL=SER=DISKXX,SPACE=(TRK,(20,,1)),DISP=(NEW,CATLG)
                (13)              (14)
```

The following points should be noted:

- AVGUTL ((8)) is the name of the new user utility. (This is the name that will occur in the automatic sequence.)
- AVRAGE ((9)) is the name of the main user subroutine in the program from 1 (Section 6.1.1.1.1).
- FPARM XREF ((10)) is an optional card. It causes the FORTRAN compiler to generate a cross-reference that is generally useful to the programmer.
- The converted program created in Section 6.1.1.1.1 must be included in this card deck at 11 .
- ZBXXX.MYFILE.LOAD ((12)) is the name of the user library in which the new utility is to be stored. ZBXXX is the user ID, MYFILE is a descriptive name chosen by the user to identify his library, and LOAD is a suffix that is always used when building utilities.
- DISKXX ((13)) is the ID of the disk on which the library is to be created.
- SPACE=(TRK,(20,,1)) ((14)) tells the amount of space on the disk that is to be reserved for the user's library. In this case, 20 tracks are to be used. The size of the user's library will vary depending on the number and size of utilities in it.

In this example a user library is created. If the user library already exists, the following LINK.SYSLMOD card should be used:

```
//LINK.SYSLMOD DD DSN=ZBXXX.MYFILE.LOAD,DISP=OLD,KEEP)
```

where ZBXXX.MYFILE.LOAD is the name of the existing library (created on a previous run), and the disposition card, DISP=(OLD,KEEP), indicates that the library already exists and should be kept on file after execution.

6.1.1.2 Using the Utility in a GMAS Automatic Sequence

To use the utility in an automatic sequence, the user must submit the deck presented below. Underlined items indicate the changes in the standard input (Figure 2-1) necessary when using utilities other than GMAS utilities.

```
col. 1
*
//ZBXXXTST JOB...
//*FORMAT...
//*FORMAT...
// EXEC GMAS,REGION.GO=300K      (15)
//STEPLIB DD DSN=ZBXXX.MYFILE.LOAD,DISP=SHR      (16)
//GO.DATA5 DD *
&CONTRL IFTUBE=50,IFTABL=49,IFTPRT=9,&END
&GMASEX SEQNAM='CARDS',IBATCH=1,NEWUT=1,UTNAME='AVGUTL',&END  (17)
DRIVE1
PRFCON
AVGUTL     (18)
 &AVGUT
  A=70.,80.,90.,100.,     (19)
 &END
EOF
/*
//
```

The following points should be noted:

- Since this utility is very small, it will not require the 375K bytes of core normally required for an orbit propagation. Instead, this case can be run in 300K ((15)).

- The STEPLIB card ((16)) must be included when the user is including his own utilities in the automatic sequence. ZBXXX.MYFILE.LOAD is the name of the user's library that contains the utility to be used in the automatic sequence.

- Two variables must be added to the &GMASEX card ((17)). NEWUT=1 indicates that one user utility will be included in the automatic sequence. UTNAME='AVGUTL' specifies the utility name. Up to 10 user utilities can be used in a run. For example, if three user utilities (UT1, UT2, and UT3) are used in an automatic sequence, the user sets NEWUT=3,UTNAME='UT1','UT2','UT3',.

- The user utility name, AVGUTL ((18)), is included at the proper point, and input is through NAMELIST AVGUT ((19)). The result of running this deck is the printed average of the four numbers under &AVGUT (see Figure 6-2).

6.1.2 EXAMPLE 2: UTILITY WITH DYNAMIC ARRAY INPUT/OUTPUT USING ROUTINES COMPUT, FECHDA, AND STORDA

In GMAS, results from one utility can be stored in a dynamic array and passed into another utility as input. This utility can process the information and store the results in a dynamic array, which can be passed in turn to another utility as the process continues. Dynamic arrays can be passed through the argument list (see Sections 6.1.3 and 6.1.4) or, as shown in this example, by using GMAS system routines COMPUT, FECHDA, and STORDA.

6.1.2.1 Creating the Utility

The objective in this example is to create a utility named AVGUTL that takes four R*8 numbers from dynamic array NUMBRS and puts their average in dynamic array AVERAG using routines COMPUT, FECHDA, and STORDA. The steps necessary to accomplish this are described below.

6.1.2.1.1 Writing the FORTRAN Program

The following FORTRAN program can be used to meet the objective in this example:

```
col. 7
*
      SUBROUTINE AVRAGE(IFRN,IERR)
```

THE AVERAGE OF 70. 80. 90. .10.3403 IS 85.

Figure 6-2. Example 1 Results

```
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(4)
      CALL COMPUT('NUMBRS',IADR1,'AVERAG',IADR2)     (1)
  IF (IADR1.EQ.0.OR.IADR2.EQ.0) GO TO 990     (2)
      CALL FECHDA(A,IADR1,4)     (3)
      SUM=0.0
      DO 100 I=1,4
           SUM=SUM+A(I)
  100 CONTINUE
      AVG=SUM/4.0
      CALL STORDA(IADR2,AVG,1)     (4)
      GO TO 1000
  990 CONTINUE
      IERR=1     (5)
      WRITE(6,995)     (6)
  995 FORMAT('1**ERROR NECESSARY ARRAYS HAVE NOT BEEN
     * ALLOCATED**')
 1000 RETURN
      END
```

The following points should be noted:

- The first three lines of code are the same as in example 1 (Section 6.1.1.1.1), except the NAMELIST statement is not used since there is no NAMELIST input in this case.

- GMAS executive service routine COMPUT is called to compute the addresses of dynamic arrays NUMBRS and AVERAG (①). A call to COMPUT must precede calls to FECHDA or STORDA since these routines require the addresses of the dynamic arrays, which are calculated in COMPUT. (Appendix E of the User's Guide provides a complete description of COMPUT.) If a dynamic array is not allocated, its corresponding address (in this case, IADR1 or IADR2) is set to 0 by COMPUT. If this occurs (②), execution continues at 990, where the error flag is set to a nonzero number (⑤) and an error message is written (⑥). Since IERR=1, the error will be captured by GMAS, thereby saving a possible system error.

- FECHDA is called (③) to transfer the contents of dynamic array NUMBRS (at address IADR1) to array A within the program. (Appendix E of the User's Guide provides a complete description of FECHDA.)

- The average is computed just as it is in example 1 (Section 6.1.1.1.1). Routine STORDA is then called to transfer the average from local variable AVG to dynamic array AVERAG (at address IADR2). Execution continues at 1000, where the RETURN statement is encountered.

6.1.2.1.2 Collecting the Necessary Information

The information necessary to create and store the utility in this example is the same as that specified in example 1 (Section 6.1.1.1.2).

6.1.2.1.3 Creating the Input Deck

The input card deck to build the desired utility in this example is the same as that specified in example 1 (Section 6.1.1.1.3).

6.1.2.2 Using the Utility in a GMAS Automatic Sequence

To use this utility in an automatic sequence, the user can submit the same card deck as specified in example 1 (Section 6.1.1.2), except the automatic sequence must be as follows:

```
col. 1
DRIVE1
PRFCON
AVGUTL
*NUMBRS,A,R8,1,4
 &VALUE
  D=70.,80.,90.,100.,
 &END
*AVERAG,A,R8,1,1
PRTDYN
 &PRTDY
  NARRS=2,
  ARRNAM='NUMBRS','AVERAG',
  ARRTIT='NUMBERS','AVERAGE',
  OUTFMT='F10.2','F10.2',
 &END
EOF
```

In this case dynamic arrays NUMBRS and AVERAG must be allocated. NUMBRS was initialized with the numbers to be averaged. After the AVGUTL utility has been executed, PRTDYN is called to print the values of the dynamic arrays. Table D-6 of the User's Guide provides a description of the NAMELIST PRTDYN input variables.

Figure 6-3 shows example 2 results.

6.1.3 EXAMPLE 3: UTILITY WITH DYNAMIC ARRAY INPUT/OUTPUT USING THE SUBROUTINE ARGUMENT LIST

Example 2 (Section 6.1.2) presents a method for accessing dynamic arrays using executive routines COMPUT, FECHDA, and STORDA. This example (example 3) illustrates another method for transferring dynamic array information into the utility.

6.1.3.1 Creating the Utility

The objective in this example is to create a utility named AVGUTL that takes four R*8 numbers from dynamic array NUMBRS and puts their average in dynamic array AVERAG using the subroutine argument list. The steps necessary to accomplish this are described below.

6.1.3.1.1 Writing the FORTRAN Program

The following FORTRAN program can be used to meet the objective in this example:

```
      col. 7
      *
      SUBROUTINE AVRAGE(NUMBRS,AVERAG,IFRN,IERR)    (1)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/DIMCOM/I1,J1,K1,I2,J2,K2    (2)
      DIMENSION NUMBRS(I1)    (3)
      SUM=0.0
      DO 100 I=1,4    (4)
           SUM=SUM+NUMBRS(I)    (5)
  100 CONTINUE
      AVERAG=SUM/4.0    (6)
      RETURN
      END
```

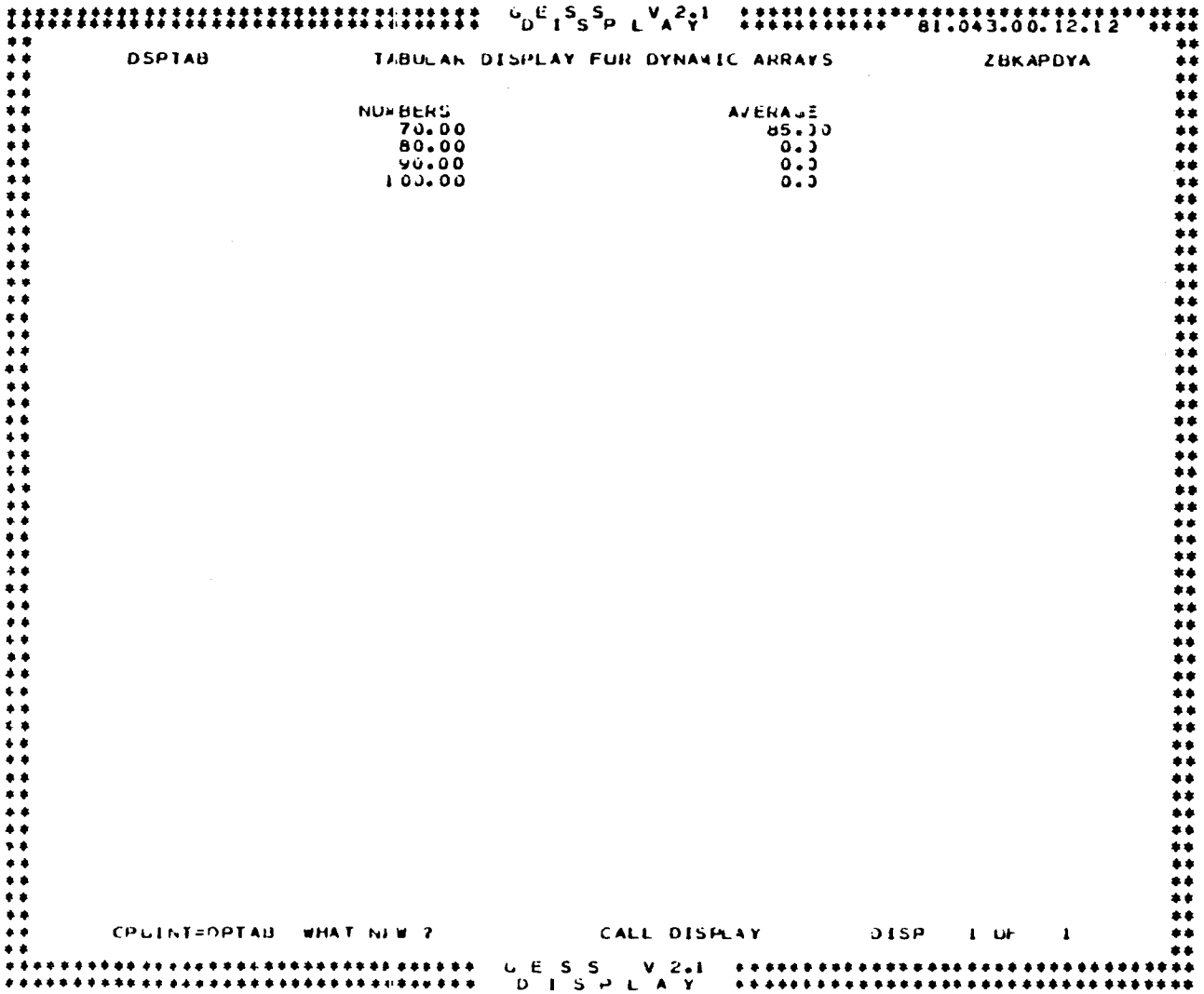


Figure 6-3. Example 2 Results

The following points should be noted:

- In this example, the dynamic arrays are passed through the argument list (①). The format of the main subroutine is as follows:

SUBROUTINE utility-main(DA_1,DA_2,...,DA_n,IFRN,IERR)

where DA_1,...,DA_n are the dynamic arrays needed by the subroutine. These dynamic arrays have been given dimensions previously by their allocation cards in the automatic sequences. This dimension information is then passed into the utility using COMMON block DIMCON (②). Therefore, the user can provide utilities that operate on dynamic arrays whose dimensions can be determined in the automatic sequence during their allocation. For example, if the user desires a utility that averages all of the numbers in a dynamic array (of arbitrary length), this can be accomplished by simply changing line ④ to DO 100 I=1,I1, and line ⑥ to AVRAG=SUM/I1. This illustrates the primary advantage in using the argument list method as opposed to the method used in example 2.

- All three dimensions must be included in COMMON block DIMCOM for each dynamic array, even if the dynamic array was allocated in one dimension. In this example, the values of I1, J1, K1, I2, J2, K2 will be 4, 1, 1, 1, 1, 1, respectively.

Page 7-1 of the User's Guide provides a complete description of passing dynamic arrays and their dimensions into a utility.

6.1.3.1.2 Collecting the Necessary Information

The information necessary to create and store the utility in this example is the same as that specified in example 1 (Section 6.1.1.1.2).

6.1.3.1.3 Creating the Input Deck

The input card deck to build the desired utility in this example is the same as that specified in example 1 (Section 6.1.1.1.3), except it must include the variables ARRNAM='NUMBRS','AVERAG', under NAMELIST UTIL to designate the existence of dynamic arrays in the utility argument list.

6.1.3.2 Using the Utility in a GMAS Automatic Sequence

The input deck setup and automatic sequence in this example are identical to those specified in example 2 (Section 6.1.2.2). The results in both examples are also identical.

6.1.4 EXAMPLE 4: UTILITY WITH DYNAMIC ARRAY INPUT/OUTPUT USING THE ARG CARD

In examples 2 and 3, the names of the input and output dynamic arrays are hard coded in the utility; input must come through dynamic array NUMBRS, and output must be transferred through dynamic array AVERAG. If, for instance, the user wants to average the numbers in another four-element array named NUM2 and put their average in an array named AVG2, he cannot use the same utility (AVGUTL). This can present quite a problem if the user has a number of four-element arrays (NUM1,NUM2,...,NUMN) from which he wants to obtain averages (AVG1,AVG2,..,AVGN) in the same automatic sequence. To do this he either needs N different utilities or must do a great deal of transferring of numbers from one array to another.

This problem can be solved using the ARG card. The ARG card, which follows the utility card in an automatic sequence, is used to specify the input/output dynamic arrays that are to be processed by the utility. For example, if the user has a utility, UTIL, that uses the ARG card to find the average of numbers in a four-element array, the

ARG card could be used as follows to accomplish the required task:

```
  :
UTIL
ARG NUM1,AVG1
UTIL
ARG NUM2,AVG2
  :
UTIL
ARG NUMN,AVGN
```

Example 4, presented below, is provided to clarify the use of the ARG card.

6.1.4.1 Creating the Utility

The objective in this example is to create a utility that adds two dynamic variables and places their sum in a third dynamic variable. All three of these dynamic variables are to be passed through the argument list using the ARG card. This utility is then to be used in an automatic sequence to add three dynamic variables by its successive execution. The steps necessary to accomplish this are described below.

6.1.4.1.1 Writing the FORTRAN Program

The following FORTRAN program can be used to meet the objective in this example:

```
col. 7
*
SUBROUTINE DYNSUM(A,B,SUM,IFRN,IERR)
IMPLICIT REAL*8(A-H,O-Z)
SUM=A+B
RETURN
END
```

The following points should be noted:

- A and B are the numbers to be input in the utility; their sum will be output in the SUM position.

- Since A, B, and SUM are scalars, DIMCOM does not need to be included.
- As usual, IFRN and IERR must be included at the end of the argument list.

6.1.4.1.2 Collecting the Necessary Information

The information necessary to create and store the utility in this example is as follows:

Information Item	Specific
Deck of cards consisting of FORTRAN code for utility	FORTRAN source deck
Name of main subroutines in source deck	DYNSUM
Name of new utility	SUMUTL
Name of library in which utility is to be stored	ZBXXX.MYFILE.LOAD
Name of disk on which library resides	DISKXX

6.1.4.1.3 Creating the Input Deck

The input card deck to build the desired utility in this example is the same as that specified in example 1 (Section 6.1.1.1.3), except UTLNAM='SUMUTL' and PRGNAM='DYNSUM'.

6.1.4.2 Using the Utility in a GMAS Automatic Sequence

To use this utility in an automatic sequence, the user can enter the same card deck as specified in example 1 (Section 6.1.1.2), except UTNAME='SUMUTL'. The automatic sequence used to add three dynamic variables using the SUMUTL utility is as follows:

Automatic Sequence	Explanation
DRIVE1 PRFCON *X,A,RB,1,1 &VALUE D=5., &END	Allocate and initialize dynamic variables

Automatic Sequence	Explanation

```
*Y,A,R8,1,1
 &VALUE
  D=10.,
 &END
*Z,A,R8,1,1
 &VALUE
  D=15.,
 &END
*SUM,A,R8,1,1
SUMUTL                                  X + Y = SUM
ARG X,Y,SUM
SUMTUL                                  Z + SUM = SUM
ARG Z,SUM,SUM
PRTDYN
 &PRTDY                                 Print values of X,
  NARRS=4,                              Y, Z and their sum
  ARRNAM='X','Y','Z','SUM',
  ARRTIT='X','Y','Z','SUM',
  OUTFMT=4*'F6.2',
 &END
EOF
```

The following points should be noted:

- The SUMUTL utility card must be followed by an ARG card with three dynamic array names.
- There must be a space between ARG and the first dynamic array.
- Although the dynamic array names are not fixed, the number and type of the dynamic arrays are not optional.

Figure 6-4 shows example 4 results.

6.2 MODIFICATION OF EXISTING UTILITIES

It is often desirable to make modifications to subroutines in a preexisting utility. This can be accomplished by using the UTLBLD program discussed in Section 6.1. To modify a preexisting module (for example, the utility module

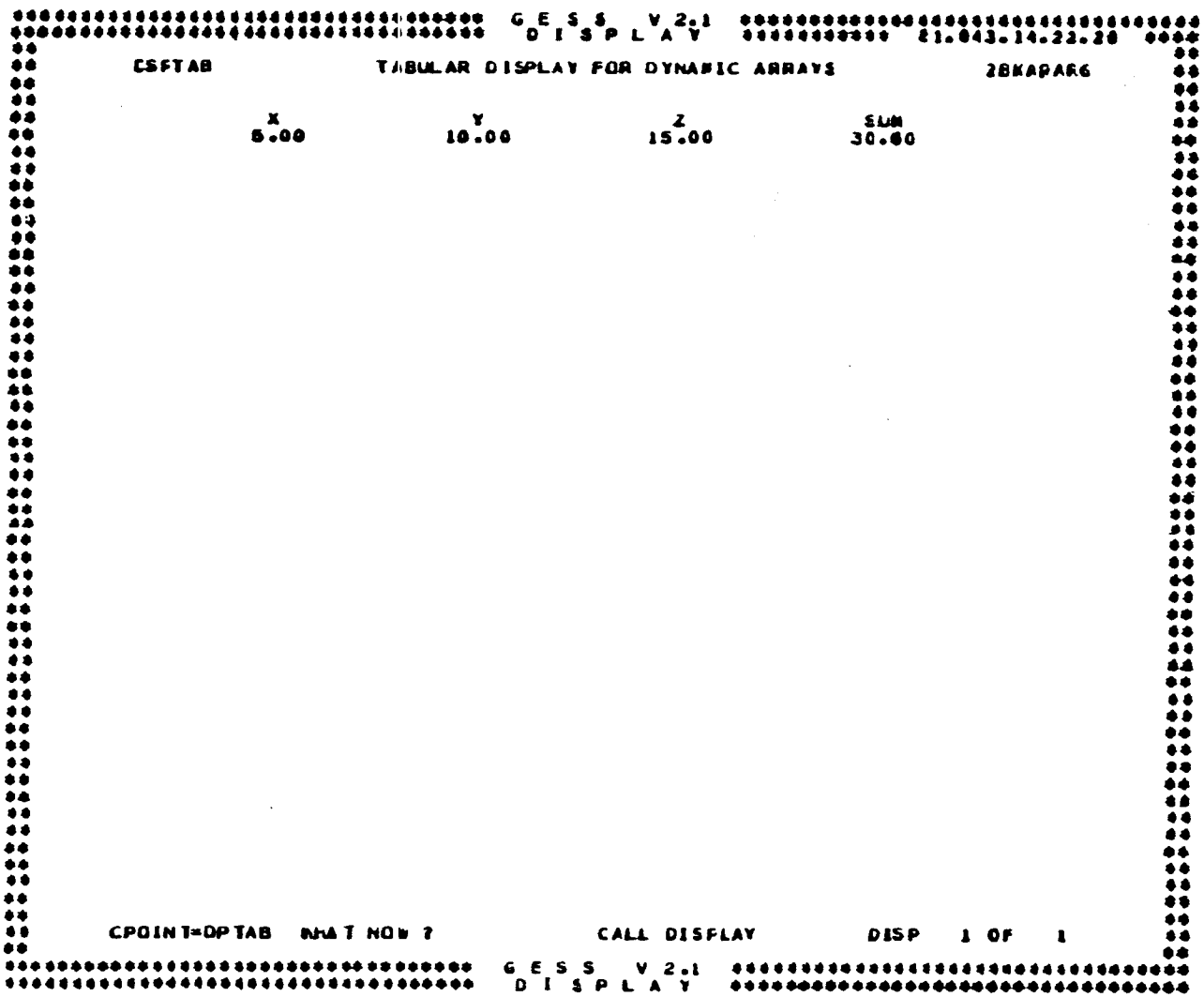


Figure 6-4. Example 4 Results

created in example 1 (Section 6.1.1)), the user should proceed as follows:

1. Collect the following information necessary to modify the preexisting module:

Information Item	Specific
Modified subroutines or subroutines to be added	AVRAGE
Name of utility to be modified	AVGUTL
Name of library containing utility to be modified	ZBXXX.MYFILE.LOAD

2. Submit the deck specified below. Underlined items indicate user-supplied information.

```
col. 1
*
//userid JOB...
//*FORMAT...
//*FORMAT...
// EXEC UTLBLD,REGION.PREP=50K     (1)
//PREP.DATA5 DD *
 &UTIL
  IOPT=1,     (2)
  UTLNAM='AVGUTL',     (3)
 &END
FPARM XREF
/*
//SOURCE.FORTIN DD *
(corrected subroutine)     (4)
/*
//LINK.SYSLIB DD DSN=ZBXXX.MYFILE.LOAD,DISP=SHR     (5)
//LINK.SYSLMOD DD DSN=ZBXXX.MYFILE.LOAD,DISP=(OLD,KEEP)     (6)
// EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//INOUT5 DD DSN=ZBXXX.MYFILE.LOAD,DISP=(OLD,KEEP)     (7)
//SYSUT3 DD UNIT=DISK,SPACE=(TRK,(1))
//SYSUT4 DD UNIT=DISK,SPACE=(TRK,(1))
//SYSIN DD *
COPYOPER COPY OUTDD=INOUT5,INDD=INOUT5
/*
//
```

The following points should be noted:

- REGION.PREP=50K (①) must be included to avoid job termination if the user is modifying an existing utility.
- IOPT=1 (②) is the switch that indicates that the user is modifying an existing utility.
- AVGUTL (③) is the name of the utility to be modified.
- The corrected subroutine(s) should be inserted at ④.
- ZBXXX.MYFILE.LOAD (⑤) is the name of the user library that contains the utility to be modified.
- ZBXXX.MYFILE.LOAD (⑥) is the name of the library that will contain the new modified utility. (Usually, the utility is put back in the same library.)
- To avoid depletion of available space in the library, it is necessary to compress the library after the modified utility has been added (⑦).

6.3 CREATION OF SPECIAL OUTPUT PARAMETER MODULES

Some mission-specific problems require output that cannot be acquired using any of the standard GMAS output values (NOUT=-3 through 3) or user-selected GMAS output values (NOUT=4 through 6). The GMAS special output (SPOUT) capability is very effective in such cases. This capability enables the user to write his own FORTRAN program (in which SPOUT must be the main routine) to be incorporated with the existing general parameter (GPARM) output module. The user's program will then be called after the last line of GPARM output (determined by NOUT). Various levels of parameters are available to the SPOUT routine through its argument list.

To use the GMAS SPOUT capability, the user should proceed as follows:

1. Write the SPOUT routine to be linked with the existing GPARM output module.
2. Use UTLBLD to create a special output parameter module that incorporates the new SPOUT routine.

Section 6.3.1 discusses the two steps specified above. Section 6.3.2 discusses using the newly created special output parameter module in a GMAS automatic sequence.

6.3.1 CREATING THE SPOUT ROUTINE AND THE SPECIAL OUTPUT PARAMETER MODULE

6.3.1.1 Writing the SPOUT Routine

The user must write the SPOUT routine in the form specified below. Underlined items indicate user-supplied information.

```
      col. 7
      ↓
      SUBROUTINE SPOUT(POS,VEL,SF,OUTPRM,TIME,ITIME,IPARMS,
     *                 NOUT,ITYPE,LNECNT,LEVEL,IERR)     (1)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION POS(3),VEL(3),OUTPRM(55),TIME(4),ITIME(8),
     *     NOUT(3)
      IF (LEVEL.EQ.0) GO TO 900     (2)
                    :
                    :
      (user's code)
                    :
                    :
      LNECNT=LNECNT+n     (3)
  900 CONTINUE
      LEVEL=m     (4)
      RETURN
      END
```

The following points should be noted:

- The argument list ((1)) contains a number of parameters that can be used in the user code. Some of the more useful parameters are discussed below.

- OUTPRM is a 55-element array that is passed through the argument list. OUTPRM contains information in the order specified in Table C-6c of the User's Guide (e.g., OUTPRM(2) = eccentricity).

- LNECNT is a variable that keeps a running total of the number of lines printed out. It is used in paginating the GMAS output parameter report. This variable must be incremented in the SPOUT routine by the number of printed lines produced in the routine (③).

- LEVEL is a variable that must be set to 1, 2, or 3, depending on the level of information the user desires passed through the OUTPRM array into the SPOUT routine. The different levels and the corresponding IPARMS values (see Table C-6c of the User's Guide) that will be computed are as follows:

Level	IPARMS Values
1	1-6
2	1-36
3	1-55

If, for example, the user needs the value of the spacecraft spin axis (IPARMS=38), he sets LEVEL=3 to ensure that the necessary computations will be carried out. The value of the spacecraft spin axis is then located in OUTPRM(38). (If the user were to set LEVEL=2, the OUTPRM(38) location might contain a meaningless value.) If, for another example, the user needs only eccentricity (OUTPRM(2)), he sets LEVEL=1.

- IERR is used as an error flag. If the user desires to flag an error in his program (e.g., to avoid division by zero), he should include an error message (using a FORTRAN WRITE statement) and set IERR=1. This will result in a GMAS error traceback printout.

- The IF test (2) must occur as the first executable line of code in the SPOUT routine. This is necessary for the proper interfacing of the SPOUT routine and the main subroutine within the GPARM module. GPARM calls SPOUT once per orbit propagation with LEVEL=0 to determine the propagation level of computation. On all other calls LEVEL is as set by the user. The user can insert his code after the IF test described above. His code can include his own subroutines or GMAS service routines (see Section 6.5 of this document and Appendixes E and F of the User's Guide).

- After the user's code LNECNT must be incremented by the number of lines (n) that were printed as a result of his code (3).

- The LEVEL indicator (4) must be set after the 900 label and before returning (m represents the level of computation necessary to make certain propagated parameters available in the OUTPRM array).

6.3.1.2 Creating the Special Output Parameter Module

To create a special output parameter module that incorporates the new SPOUT routine and stores it in the user's library, the user must submit the card deck specified below. Underlined items indicate user-supplied information.

```
col. 1
*
//userid JOB...
//*FORMAT...
//*FORMAT...
// EXEC UTLBLD,REGION.PREP=50K,PARM.LINK='MAP,LIST,LET,OVLY'
//PREP.DATA5 DD *
 &UTIL
  IOPT=1,
  UTLNAM='name',     (1)
  OLDUTL='GPARM',OLDENT='IMPLS',
 &END
FPARM XREF
```

```
//SOURCE.FORTIN DD *
        :
      (SPOUT routine)    (2)
        :
//PREP.LINKIN DD DSN=GJMAS.OVERLAY.DATA(GPARM),
//      DISP=SHR,LABEL=(,,,IN)
//LINK.SYSLIB DD DSN=GJMAS.GO.LOAD,DISP=SHR
//LINK.SYSLMOD DD DSN=userid.filename.LOAD,UNIT=DISK,    (3)
//      VOL=SER=DISKxx,SPACE=(TRK,(20,,1)),DISP=(NEW,CATLG)    (4)
```

The user must include the name of the new output parameter module ((1)), the SPOUT source code (see Section 6.3.1.1) ((2)), and the names of the user's library ((3)) and the disk ((4)).

6.3.2 USING THE SPECIAL OUTPUT PARAMETER MODULE IN A GMAS AUTOMATIC SEQUENCE

To include the special output parameter module in an automatic sequence, the user must submit a deck in the form specified below. Underlined items indicate user-supplied information.

```
col. 1
▼
//userid JOB...
//*FORMAT...
//*FORMAT...
// EXEC GMAS,REGION.GO=400K
//STEPLIB DD DSN=userid.filename.LOAD,DISP=SHR    (5)
//GO.DATA5 DD *
 &CONTRL IFTUBE=50,IFTABL=49,IFTPRT=9,&END
 &GMASEX SEQNAM='CARDS',IBATCH=1,&END
DRIVE1
PRFCON
ORBINP
 &ORBINP
      :
  (orbital input information)
      :
  PARMM='name'    (6)
 &END
ORBIT
EOF
```

The following points should be noted:

- A STEPLIB card must be included with the name of the user's library (see Section 6.3.1.2). (⑤ and ③ (from Section 6.3.1.2) should be the same.)
- The automatic sequence can be written in the normal fashion, except the user should substitute his special output parameter module for the GPARM module. This is done using variable PARMM. (⑥ and ① (from Section 6.3.1.2) should be the same.)

6.4 CREATION OF PARAMETER MODULES

In Section 6.3, the general parameter (GPARM) output module is replaced by the user's special output module, which is the same as GPARM except that it contains a user-defined SPOUT routine that causes special user-defined output to be printed at each stop after the standard GMAS output. Modules such as this one are called parameter modules. Parameter modules are like utilities in that they are self-contained programs (load modules) that can be built by using the UTLBLD procedure. They differ from utilities in the following ways:

- Parameter modules are not placed in the automatic sequence. They are specified through variable PARMM (PARMM='GPARM') in NAMELIST ORBIN.
- To use a new utility in an automatic sequence, the user must include variables NEWUT and UTNAME in NAMELIST GMASEX. This is not necessary when using a new parameter module.
- Input to utilities can be through NAMELIST variables or dynamic arrays. Input to parameter modules is usually through a fixed argument list.
- The parameter module is executed each time the orbit propagator achieves a stopping condition.

In general, a parameter module should be used instead of a utility module if the user wants to execute the module at specific events during the propagation of a satellite orbit. However, a parameter module can be executed without the propagation of an orbit. This can be accomplished by including the variable value IPOPT=99 under NAMELIST ORBIN.

To create a parameter module from a subroutine, existing program, or logically connected group of subroutines, the user should proceed as follows:

1. Convert the main routine to a subroutine.
2. Use UTLBLD to create the new parameter module and store it in the user's library.

Section 6.4.1 discusses the two steps specified above. Section 6.4.2 discusses using the parameter module in a GMAS automatic sequence.

6.4.1 CONSTRUCTING THE PARAMETER MODULE

6.4.1.1 Constructing the Main Routine of the Parameter Module

To construct the main routine of a parameter module, the user must write the routine in the form specified below. Underlined items indicate user-supplied information.

```
col. 7
▼
SUBROUTINE param(STATEI,STATEF,PARMS,NOUT,ITYPE,    (1)
                 IDYN,IPARMS,IERR)
IMPLICIT REAL*8(A-H, O-Z)
COMMON/DIMCOM/I1,J1,K1,I2,J2,K2,I3,J3,K3
DIMENSION STATEI(I1),STATEF(I2),PARMS(I3,J3),NOUT(3),
*     IPARMS(10)
      :
      :
(user code)
      :
      :
RETURN
END
```

The following points should be noted:

- Entry param (①) is the user-supplied name of the main routine in the program that is to be converted into a parameter module. Parameter modules have a fixed argument list. STATEI, STATEF, and PARMS are dynamic arrays, and the other variables are passed through the argument list (like IFRN and IERR in the main routine of a utility; see Section 6.1.3). Section 3.3.4.2.1 of the GMAS System Description provides a complete description of the variables in the argument list.

- NOUT, IDYN, and IPARMS contain the values input under NAMELIST ORBIN in the automatic sequence preceding the ORBIT card. These variables are often used to pass information into the parameter module. An example of this is COMPLM (a parameter load module used to calculate various parameter differences of two orbits), which uses nonstandard values of the variables as input. (See Section C.2.5 of the User's Guide.)

- ITYP is an indicator flag having the following values:

Value	Meaning
-1	Initial entry to the parameter module (before a call is made to the propagator)
0	Intermediate call to the parameter module (after a stopping condition has been met)
1	Final call to the parameter module

If ITYP is set to 2 by the user, ORBIT propagation is terminated.

- IERR is the error condition indicator. If an error occurs during the execution of the user's parameter module, IERR should be set to a nonzero value.

6.4.1.2 Creating the Parameter Module

To create the new parameter module and store it in the user's library, the user must submit a card deck in the form specified below. Underlined items indicate user-supplied information.

```
col. 1
*
//userid JOB...
//*FORMAT...
//*FORMAT...
// EXEC UTLBLD
//PREP.DATA5 DD *
 &UTIL
  UTLNAM='PRMOUT',PRGNAM='parm',   (2)   (3)
  ARRNAM='STATEI','STATEF','PARMS',
 &END
FPARM XREF
/*
//SOURCE.FORTIN DD  *
      :
      :
(source code)
      :
      :
/*
//LINK.SYSLMOD DD DNS=ZBXXX.MYFILE.LOAD,UNIT=DISK,     (4)
//      VOL=SER=DISKxx,SPACE=(TRK,(20,,1)),    (5)
//      DISP=(NEW,CATLG)
```

The following points should be noted:

- PRMOUT ((2)) is the user's parameter module name (the user sets PARM='PRMOUT' in NAMELIST ORBIN).
- Entry parm ((3)) is the name of the main routine in the FORTRAN program (see Section 6.4.1). ((1) and (3) should be the same.)
- (4) is the name of the user's library, and (5) is the disk on which it will reside.

6.4.2 USING THE PARAMETER MODULE IN A GMAS AUTOMATIC SEQUENCE

To use the new parameter module in an automatic sequence, the user must proceed as specified in Section 6.3.2.

6.4.3 MISCELLANEOUS ASPECTS OF USER MODULE CREATION

The following points should be noted concerning the creation of user modules:

- The UTLBLD deck setups given in this section and Section 6.3 are to be used if the user is creating a library. If a library already exists, or if the user wants to update an existing parameter module or special output routine, he should use the method described in Section 6.2 and refer to Section 7 of the User's Guide. If the user's library contains only one utility or parameter module, it may be easier to delete the entire library and start over again. This can be accomplished by putting the following two cards after the //*FORMAT cards in the UTLBLD deck:

```
col. 1
*
// EXEC PGM=IEFBR14
//DELETE DD DSN=ZBxxx.MYFILE.LOAD,DISP=(OLD,DELETE)
```

where ZBxxx.MYFILE.LOAD is the library file to be deleted.

- The user can also modify or create propagators. Such modules, called propagator modules, are specified through variable PROPM (PROPM='TRCOWL') in NAMELIST ORBIN. The building and use of propagator modules is similar to that of parameter modules, except the propagator module main subroutine has a different argument list than the parameter module main subroutine.

6.5 USE OF GMAS SERVICE ROUTINES IN USER LOAD MODULES

A number of GMAS service routines can be called from a user FORTRAN program that is to become a utility or a parameter module. Most service routines can be classified into the following categories:

- Dynamic array handling routines
- Coordinate and element conversion routines

- Matrix and vector manipulation routines
- Time and date conversion routines

Other service routines compute the state transition matrix (TRANMX), compute the partials of the Keplerian elements with respect to the Cartesian elements (KPART), and perform many other useful services. Appendixes E and F of the User's Guide provide a complete categorized list of all available service routines.

The use of service routines is illustrated in the following example. The objective in this case is to write a FORTRAN program to be converted into a GMAS utility that calculates the magnitude of the position and velocity vectors of a satellite, given the Keplerian elements through dynamic array KEPLER and the gravitational constant times the mass of the central body through dynamic array GMC. The dynamic arrays involved are as follows:

Dynamic Arrays	I/O	Dimension
KEPLER	I	6
GMC	I	1
POSMAG	O	1
VELMAG	O	1

The solution program for this problem is specified below. Underlined routines indicate GMAS service routines used.

```
SUBROUTINE MAG(KEPLER,GMC,POSMAG,VELMAG,IFRN,IERR)
COMMON/DIMCOM/I1,J1,K1,I2,J2,K2,I3,J3,K3,I4,J4,K4)
DIMENSION KEPLER(I1),GMC(I2),POSMAG(I3),VELMAG(I4)
DIMENSION PV(3),VV(3)
CALL CELEM(KEPLER,GMC,PV,VV)
POSMAG=SQRT(FDOT(PV,PV,3))
VELMAG=SQRT(FDOT(VV,VV,3))
RETURN
END
```

SECTION 7 - MISCELLANEOUS GMAS CAPABILITIES

Since this document is a primer, it does not cover all aspects or capabilities of GMAS and its connected software. This section introduces some of the GMAS capabilities not covered in the primer and specifies appropriate sources of reference for these capabilities for interested readers.

7.1 GMAS INTERACTIVE MODE

Although this document deals only with GMAS in the batch mode (i.e., input via card deck), GMAS can also be run in an interactive mode using one of the GSFC cathode ray tube (CRT) graphics terminals (IBM 2250, IBM 2260, Anagraph 6600). In the interactive mode, automatic sequences can be edited and created (automatic mode), or utilities can be executed individually at the user's discretion (manual mode). Section 4 of the User's Guide presents a complete description of GMAS interactive capabilities and operating instructions.

7.2 GMAS AUTOMATIC SEQUENCE LIBRARIES

GMAS has a library of preexisting automatic sequences. These automatic sequences can be accessed by replacing SEQNAM ='CARDS' with SEQNAM ='automatic sequence name' in NAMELIST GMASEX.

In Section 5.6 of this document, the preexisting automatic sequence MEANEL is used to create mean elements to be input in the averaged orbit propagator. The user can develop his own library of automatic sequences to complement those existing in the GMAS library. The user must include the card

```
//FT04F001 DD DSN = 'userid.userfilename.DATA',DISP=SHR
```

where the user's file is a partitioned data set (PDS) whose members are automatic sequences (sequential data sets of record length 80).

Preexisting automatic sequences are of very little value unless the user has a method of updating them. Section 3.3 of the User's Guide describes the GMAS automatic sequence updating procedure.

7.3 GESS EXECUTIVE

The Graphic Executive Support System (GESS) is a large system that provides not only an interactive graphics capability, but also a variety of user services. Reference 7 provides a detailed description of GESS.

7.4 IOHAND

IOHAND is an input/output handler program originally developed for the Graphics Mission Operations Support System (GMOSS). Although it is not a GMAS utility, IOHAND may be useful to GMAS users performing online support functions. Section 4.1 of the Software Resources document provides a complete description of IOHAND.

REFERENCES

1. Computer Sciences Corporation, CSC/SD-80/6028, Goddard Mission Analysis System (GMAS) System Description, G. A. Snyder and E. J. Smith, June 1980

2. --, CSC/SD-79/6059, Goddard Mission Analysis System (GMAS) User's Guide, G. A. Snyder and E. J. Smith, May 1979

3. --, CSC/SD-79/6079, Software Resources for Use With the Goddard Mission Analysis System (GMAS), G. A. Snyder, October 1979

4. --, CSC/SD-79/6079UD1, Updates to Software Resources for Use With the Goddard Mission Analysis System (GMAS), G. A. Snyder, March 1980

5. --, CSC/SD-79/6079UD2, Updates to Software Resources for Use With the Goddard Mission Analysis System (GMAS), D. C. Folta and W. T. Wallace, November 1980

6. International Business Machines Corporation, GC28-6515-9, IBM System/360 and System/370 FORTRAN IV Language, October 1972

7. Computer Sciences Corporation, CSC/SD-75/6057, Graphic Executive Support System (GESS) User's Guide, J. Hoover et al., August 1975